11-02-00 A

SKJERVEN
MORRILL
MACPHERSON LLP

JC962 U.S. PTO
10/31/00

Docket No.: M-9582 US

JC922 U.S. PTO
09/703426
10/31/00

October 31, 2000

Box Patent Application
Commissioner For Patents
Washington, D. C. 20231

Enclosed herewith for filing is a patent application, as follows:

Inventor(s): Rash, Clarence E.; Hsieh, Sheng-Jen; Harding, Thomas H.; Beasley, Howard H.; Martin, John S.; Reynolds, Ronald W.; Dillard, Robert M.

Title: Image Quality Tester

X	Return Receipt Postcard
X	This Transmittal Letter (in duplicate)
106	page(s) Specification (not including claims)
7	page(s) Claims
1	page Abstract
48	Sheet(s) of Drawings
3	page(s) Declaration For Patent Application and Power of Attorney (unsigned)
2	Information Disclosure Statement Under 37 C.F.R. § 1.97(b) (2 pgs.)
1	page(s) PTO Form 1449 citing 1 reference
X	Copy of 1 Cited Reference submitted

CLAIMS AS FILED

For	Number Filed			Number Extra		Rate			Basic Fee 710.00
Total Claims	24	-20	=	4	x	$ 18.00	=	$	72.00
Independent Claims	3	-3	=	0	x	$80	=	$	0.00
☐ Fee of ______ for the first filing of one or more multiple dependent claims per application								$	
☐ Fee for Request for Extension of Time								$	

Please make the following charges to Deposit Account 19-2386:

☒ Total fee for filing the patent application in the amount of $ 782.00

☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account 19-2386.

EXPRESS MAIL LABEL NO:

EL708268375US

Respectfully submitted,

Dale R. Cook
Attorney for Applicants
Reg. No. 42,434

"Express Mail" mailing label number:

EL708268375US

IMAGE QUALITY TESTER

Clarence E. Rash
Sheng-Jen Hsieh
Thomas H. Harding
Howard H. Beasley
John S. Martin
Ronald W. Reynolds
Robert M. Dillard

CROSS REFERENCED TO RELATED APPLICATION

The present invention is related to the subject matter of the provisional United States Patent Application entitled, "Field-Ready Monocular Helmet Mounted Display Imagery Evaluation System," naming inventors Clarence E. Rash, Thomas H. Harding, Sheng-Jen Hsieh, Howard H. Beasley, John S. Martin, Ronald W. Reynolds, and Robert M. Dillard, filed October 11, 2000, Attorney Docket No. M-9582 V1. Applicants hereby claim the benefit under 35 U.S.C. §119(e) of the foregoing-referenced provisional application. The contents of the foregoing-referenced provisional patent application are hereby incorporated by reference herein in its entirety.

BACKGROUND OF THE INVENTION

Field of the Invention

The present application relates, in general, to methods and systems related to testing of display sighting systems.

Description of the Related Art

A display sighting system, in general, is a system which allows a pilot to integrate himself with a vehicle he is piloting. A display sighting system will generally have one or more unique features depending upon the environment in which the display sighting system is deployed. One feature which a display sighting system might have is the capability to allow a human pilot to see beyond the limitations of normal human vision (e.g., forward looking infrared radar (FLIR) systems, which use radar, image intensification, and infrared waves (which ordinarily cannot be seen by a human pilot) to construct and project a picture which a human pilot can see). Another

feature which a display sighting system might have is the integration of the display sighting system with weapons control of a helicopter (e.g., integrated such that the visual display of the display sighting system is integrated with the gun sights of the weapons on the helicopter).

One example of a display sighting system is the Integrated Helmet and Display Sighting system (IHADSS) of the U.S. Army's AH-64 Apache helicopter. (For ease of understanding, the present discussion will refer throughout to the IHADSS, but it will be recognized by those having ordinary skill in the art that the IHADSS is intended to be representative of the more general display sighting systems referenced above.)

The IHADSS typically gathers information related to the terrain and environment in which the AH-64 is operating by using cameras and/or sensors affixed to the AH-64. Thereafter, the IHADSS processes the gathered information into a form which can be seen by a human pilot, and thereafter projects the gathered and processed information via an assortment of electronic and optical apparatuses (described in more detail, below) into a human pilot's field of view. In many instances, a pilot of an AH-64 is actually flying the helicopter or targeting the helicopter's weapons systems on the basis of what is displayed by the IHADSS. Accordingly, it is imperative that each individual IHADSS project a clear and accurate depiction of the terrains and/or environments captured by its associated cameras and sensors. Unfortunately, the integration of each individual IHADSS with the systems and subsystems of AH-64 helicopters in which each individual IHADSS is deployed makes it difficult to ensure that each individual IHADSS is projecting a clear, accurate, and quality depiction of the terrains and/or environments captured by its associated cameras and sensors. This difficulty is due in large part to a lack of effective methods and systems for the assessment of the accuracy and quality of IHADSS imagery in a field environment.

At present, when an IHADSS is deployed in a field environment, the accuracy and quality of the deployed IHADSS imagery is determined on a subjective basis by each pilot viewing the IHADSS' display device. Those having ordinary skill in the art will recognize that such a methodology is suboptimal for several reasons. One

reason why such methodology is suboptimal arises from interaction of the gradual degradation of the IHADSS with the adaptability of the human visual system. Those having ordinary skill in the art will recognize that over time it is common for the visual displays of IHADSS to gradually degrade and become distorted for various reasons (e.g., aging of the electronics, routine wear and tear, shock and vibration, etc.). It has been found by the inventors named herein ("the inventors") that in practice, an IHADSS display can be substantially degraded without such degradation being detectable by the human pilot, because insofar as each IHADSS is typically tuned by a specific human pilot, and insofar as the degradation of the IHADSS over time is often gradual, the adaptability of the human visual system often tricks the human pilot into thinking IHADSS display is accurate and/or acceptable when in fact it is substantially inaccurate and/or unacceptable. Another reason why the current methodology is suboptimal arises from the lack of accuracy and/or reproducibility generally associated with subjective approaches.

In light of the foregoing, it is clear that a need exists for a method and system which will practicably allow the objective assessment of the functioning of individual IHADSSs used by pilots in the field. Unfortunately, those skilled in the art will recognize that the objective testing of display sighting systems such as the IHADSS is generally a very involved process which at present is generally only done in the laboratory environment via use of a series of well defined discrete operations performed with separate items of test equipment. Those skilled in the art will recognize that, in general, the testing done in the laboratory environment is not adaptable to operation in the field.

One reason why the testing done in the laboratory environment is generally not adaptable to operation in the field is that once IHADSS have been deployed, operators are prohibited from changing the hardware and/or software within the IHADSS without a specific Army requirement and without cooperation from the IHADSS manufacturer. As a consequence of this, the sophisticated hardware and software often used in the laboratory environment cannot be used to test IHADSS in the field, since what can be used consists of only what is built-in to the IHADSS. Another reason why testing done in the laboratory environment is generally not adaptable to operation in the field is that the delicate testing equipment used in the lab

is generally not appropriate for field testing. Yet another reason why testing done in the laboratory environment is generally not adaptable to operation in the field is that it is generally not practicable to deploy the large and bulky testing equipment used in the lab into the field environment.

In light of the fact that there is at present no practicable way to adapt IHADSS testing done in the laboratory environment to field environments, it is therefore apparent that the need exists for a method and system which will practicably allow the objective assessment of the functioning of individual IHADSSs in a field environment.

SUMMARY OF THE INVENTION

The inventors have devised a method and system which will practicably allow the objective assessment of the functioning of individual display sighting systems, such as IHADSSs, in a field environment.

In one implementation, a method includes but is not limited to capturing an image actually displayed via a display sighting system; computing at least one difference between the captured image and a recalled representation of the image theoretically displayed via the display sighting system; and presenting the computed at least one difference via a visual display device. In various implementations, circuitry is used to effect the foregoing-described method; the circuitry can be virtually any combination of hardware, software, and/or firmware configured to effect the foregoing-described method depending upon the design choices of the system designer.

The foregoing is a summary and thus contains, by necessity, simplifications, generalizations and omissions of detail; consequently, those skilled in the art will appreciate that the summary is illustrative only and is not intended to be in any way limiting. Other aspects, inventive features, and advantages of this patent application will become apparent in the non-limiting detailed description set forth below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

Figure 1 illustrates a perspective view of pilot 100 wearing IHU 102

Figure 2 depicts a perspective view of HDU 104 in isolation.

Figures 3A, 3B, 3C, 3D and 3E show three different perspective views of positioning device 300, which is an apparatus which allows HDU 104 in holding fixture 390 to be moved between a first position (see Figure 3D) and a second position (see Figure 3E) such that an image projected by HDU 104 onto combiner 106 can be respectively captured by narrow-angle camera 306 and wide-angle camera 308 (see Figures 3D and 3E).

Figure 4 illustrates test pattern 400 which comes built-in with the IHADSS produced by Honeywell, Inc.

Figure 5 shows a high-level logic flowchart.

Figure 6A depicts alternate implementations of the high-level logic flowchart depicted in Figure 5.

Figures 6B and 6C illustrate pictographic representations of the methodology used by one implementation to compute the centering difference of method step 602 and the angular difference of method step 600.

Figure 7 illustrates alternate implementations of the high-level logic flowchart depicted in Figure 6A.

Figure 8 shows alternate implementations of the high-level logic flowchart depicted in Figure 5.

Figure 9 depicts alternate implementations of the high-level logic flowchart depicted in Figure 8.

Figure 10 illustrates a pictorial representation of a conventional data processing system in which illustrative embodiments of the devices and/or processes described herein may be implemented.

Figure A1 shows the IHADSS.

Figure A2 depicts The IHADSS HDU.

Figure A3 illustrates display size.

Figure A4 shows the built-in test pattern of the IHADSS HMD.

Figure A5 depicts a configuration in which two cameras face the HMD from different directions.

Figure A6 illustrates a revised design of that depicted in Figure A5.

Figure A7 shows locations of sensors in proposed HMD fixture design.

Figure A8 depicts a typical DAQCard-DIO-24 configuration.

Figure A9 illustrates how the HMD hardware fixtures, sensors, I/O cable, and DAQCard-DIO-24 card are integrated.

Figure A10 shows initial display screen, switches open.

Figure A11 depicts a display screen, continue button, switches open.

Figure A12 illustrates display screen, switches 1 and 2 pressed.

Figure A13 shows display screen, image capture module activated.

Figure A14 depicts a screenshot of the newly designed image capture interface module, showing an image of the HMD test pattern taken using the wide-angle camera.

Figure A15 illustrates screenshots of parameter setting display screens.

Figure A16 shows an original image/test pattern.

Figure A17 depicts the image of Figure A16 after binary processing.

Figure B1 illustrates the IHU of the AH-64 IHADSS.

Figure B2 shows the IHADSS HDU.

Figure B3 depicts the display size.

Figure B4 shows a snapshot of the test pattern captured from the IHADSS HMD.

Figure B5 illustrates a flow chart for HMD prototype tester operation.

Figure B6 depicts an experimental setup.

Figure B7 shows sampling locations on a test pattern.

Figure B8 illustrates a plot of photometer and CCD camera data.

Figure B9 shows a set up for test pattern measurement.

Figure B10 illustrates a test pattern design based on measurement results.

Figure B11 displays a replicated test pattern image.

Figure B12 shows the measurement of luminance of the center lines.

Figure B13 illustrates center lines measurement with varied focus.

Figure B14 displays a designed test pattern with focusi on the center lines.

Figure B15 shows modules involved in the prototype.

Figure B16 shows the opening screen for the image capture module.

Figure B17 depicts an image capture component.

Figure B18 illustrates an image processing component.

Figure B19 shows an image analysis and interpretation module.

Figure B20(a-d) depicts tilted test pattern binary images from image analysis module.

Figure B21 illustrates overall testing results of an HMD.

Figure B22 shows tilted test pattern before and after Sober edge detection.

Figure B23 depicts investigation of CCD image capture arrangement.

Figure B24 illustrates a preliminary computer aided design (CAD) concept of a hardware prototype design.

The use of the same reference symbols in different drawings indicates similar or identical items.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

As described above, the U.S. Army's AH-64 Apache helicopter incorporates a monocular helmet mounted display (HMD) known as the Integrated Helmet and Display Sighting System (IHADSS). The IHADSS consists of various electronic components and a helmet/display system called the Integrated Helmet Unit (IHU).

With reference to the figures, and in particular with reference now to Figure 1, shown is a perspective view of pilot 100 wearing IHU 102. Depicted is that IHU 102 includes Helmet Display Unit (HDU) 104. HDU 104 serves as an optical relay device which conveys an image formed on a mini-CRT through a series of lenses (the mini-CRT and lenses are internal to HDU 104, and hence are not shown explicitly), off beamsplitter (combiner) 106, an into pilot 100's right eye. Combiner 106 is so named in that its construction allows pilot 100 to see whatever image is projected from HDU 104 superimposed, or combined with, whatever pilot 100 can see through combiner 106; in effect, combiner 106 functions as a mirror with respect to the projection of the mini-CRT within HDU 104, and a lens with respect to allowing pilot 100 to see what is in front of him. That is, if one looks closely at Figure 1, one can see that pilot 100's eye is visible through combiner 106, which means that pilot can see through combiner 106. Consequently, what pilot 100 sees will be a combination of the projection of HDU 104 and what appears in front of pilot 100 (i.e., whatever pilot 100 can see via the lens function of combiner 106).

Referring now to Figure 2, shown is a perspective view of HDU 104 in isolation.

Referring now to Figures 3A, 3B, and 3C, shown are three different perspective views of positioning device 300, which is an apparatus which allows HDU 104 in holding fixture 390 to be moved between a first position (see Figure 3D) and a second position (see Figure 3E) such that an image projected by HDU 104 onto combiner 106 can be respectively captured by narrow-angle camera 306 and wide-angle camera 308 (see Figures 3D and 3E). Further shown in Figures 3B and 3C is spring 350 which has first end 358 in stationary attachment with camera table 360 and second end 362 in mobile attachment t-shaped-end of lever 352. Lever 352 is shown attached to pivot pole 356. A second end (not shown) of lever 352 is pivotably attached to the underside of holding fixture 390 such that when holding fixture 390 is moved back and forth along guide rails 364 and 366, the second end of lever pivotably attached to the underside of holding fixture 390 will pivot such that t-shaped-end of lever 352 will pivot about pivot pole 356.

In operation, when HDU 104 in holding fixture 390 is moved between a first position (see Figure 3D) and a second position (see Figure 3E) by sliding along guide rails 364 and 366, lever 352 pivots about pivot pole 356 and second end 362 of spring 350 slides along t-shaped-end of lever 352 and is such that when holding fixture 390 is in either the first or second position, spring 350 applies pressure through t-shaped-end of lever 352 such that pressure is applied through second end of lever 352 to holding fixture 390 such that holding fixture 390, and hence HDU 104, is held in either that first or second position.

With reference now to Figures 3D and 3E, shown are two different perspective views of HDU 104 in positioning device 300, which is an apparatus which allows HDU 104 to be moved between a first position and a second position such that an image projected by HDU 104 onto combiner 106 can be respectively captured by narrow-angle camera 306 and wide-angle camera 308.

Referring now to Figure 4, shown is test pattern 400 which comes built-in with the IHADSS produced by Honeywell, Inc. The original main purpose of test pattern 400 was to provide Honeywell field engineers with something consistent to view on

the IHADSS when the IHADSS was first deployed to the field. The original secondary purpose of test pattern 400 was to provide pilots with a fixed pattern which the pilots could use to subjectively optimize their display. Consequently, test pattern 400 is substantially always available on virtually every IHADSS systems. (It is to be understood that although the test pattern of the IHADSS is discussed herein for sake of simplicity, the discussion is meant to be representative of test patterns utilized with display sighting systems, and such test patterns can come preloaded from the factory or may be loaded after the fact, and such test patterns may be multiple in number.)

Depicted is that in one embodiment, test pattern 400 is treated as having two portions: wide-angle portion 402, which is co-extensive with test pattern 400 itself, and narrow-angle portion 404, which is denoted as the small rectangular portion substantially between the gray-scale strips in Figure 4. Wide-angle portion 402 and narrow-angle portion 404 are so named because in one embodiment wide-angle camera 308 and narrow-angle camera 306 are used to respectively capture such portions.

With reference now to Figure 5, shown is a high-level logic flowchart. Method step 500 depicts the start of the process. Method step 502 illustrates capturing an image of IHADSS test pattern 400. Method step 504 shows computing at least one difference between the captured image of IHADSS test pattern 400 and a recalled representation of the image of IHADSS test pattern 400 theoretically displayed via the display sighting system; that is, there is stored in memory (e.g., computer memory) a representation of how the image of IHADSS test pattern 400 would or should appear if the IHADSS is both functioning correctly and set up properly (e.g., by a pilot), and it is such a representation that is recalled. Method step 506 shows presenting the computed at least one difference via a visual display device (e.g., via a portable computer system display). Method step 508 shows the end of the process.

Referring now to Figure 6A, shown are alternate implementations of the high-level logic flowchart depicted in Figure 5. Method step 600 illustrates that in one implementation method step 504 includes computing at least one angular difference between an angular orientation of the captured image and the recalled representation

of the image theoretically displayed via the display sighting system. Method step 602 shows that in one implementation method step 504 includes computing at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system. Method step 604 depicts that in one implementation method step 504 includes computing at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system. The remaining method steps of the flowchart depicted in Figure 6 function substantially as shown and described herein.

In one instance, the implementations of Figure 6A are carried out by capturing narrow-angle portion 404 of IHADSS test pattern 400 with narrow-angle camera 306, converting the captured narrow-angle portion to "binary form," and recalling a representation of what narrow-angle portion 404 of IHADSS test pattern 400 should look like if the IHADSS system were functioning substantially optimally. As used herein converting to "binary form" means creating a two-color image, where all captured image pixels having a number below a calculated threshold are set to gray level 0 (pure black) and all captured image pixels above a calculated threshold value are set to gray level 255 (pure white in a system with 0-255 gray levels). Specific examples of the foregoing described conversion to binary images appear in Specific Implementation A, Design of Interface and Algorithms for an Image Quality Tester (see, especially Figures A16 and A17 and the discussion of same), and Specific Implementation B, Preliminary Design of an Image Quality Tester for Helmet-Mounted Displays.

Referring now to Figures 6B and 6C, shown are pictographic representations of the methodology used by one implementation to compute the centering difference of method step 602 and the angular difference of method step 600. Figure 6B shows that, in one implementation of method step 602, the center point of the actual image captured (point "B") can be compared with where the center point should be if the IHADSS were functioning substantially optimally (point "A," which is obtained from the recalled representation of the image theoretically displayed via the display sighting system), and the resulting x-y displacement used to compute the distance difference, d, between actual and theoretical center point locations (the x-y units can

be any unit of length, but in one embodiment the x-y units are millimeters). With reference to Figure 6C, shown is that a right triangle drawing on the x-y position relative to points A and B can be used in combination with a point chosen to be on the "vertical" line of captured narrow-angle portion 404 of IHADSS test pattern 400 in order to calculate θ (theta) as the orientation difference between the captured image orientation and the theoretically ideal images. Specific examples of the foregoing appear in Specific Implementation A, Design of Interface and Algorithms for an Image Quality Tester (especially in the Algorithm Design section, subsection C, wherein various approaches for identifying the center point and identifying test pattern orientation and displacement are described), and in Specific Implementation B, Preliminary Design of an Image Quality Tester for Helmet-Mounted Displays.

With respect to computing the focus difference of method step 604, note that narrow-angle portion 404 of IHADSS test pattern 400, subsequent to being converted to binary form, has four bright white vertical lines bounded by dark black regions. It has been discovered by the inventors that measurement of luminance differences between the alternating bright white and dark black regions can be utilized to determine focus. For example, in Specific Implementation B, Preliminary Design of an Image Quality Tester for Helmet-Mounted Displays, Figure B12 and its supporting text shows how luminance measurements appear when narrow-angle portion 404 of IHADSS test pattern 400 is in focus. In contrast, Figure B13 and its supporting text shows several examples of how luminance measurements appear when narrow-angle portion 404 of IHADSS test pattern 400 is out of focus. Consequently, in one implementation, the focus difference of method step 604 can be calculated based on how far the measured luminances of the alternating black and white lines of the binary image form of captured narrow-angle portion 404 of IHADSS test pattern 400 varies from the ideal (e.g., such as that appearing in Figure B12).

With reference now to Figure 7, depicted are alternate implementations of the high-level logic flowchart depicted in Figure 6A. Method step 700 illustrates that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer) the at least one angular difference between an angular orientation of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of

method step 600). Method step 702 shows that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 602). Method step 704 depicts that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 604). The remaining method steps of the flowchart depicted in Figure 7 function substantially as shown and described herein.

Referring now to Figure 8, shown are alternate implementations of the high-level logic flowchart depicted in Figure 5. Method step 800 illustrates that in one implementation method step 504 includes computing at least one gray-shades-displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system. Method step 802 shows that in one implementation method step 504 includes computing at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system. Method step 804 shows that in one implementation method step 504 includes computing at least one image quality figure of merit indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system. The remaining method steps of the flowchart depicted in Figure 8 function substantially as shown and described herein.

In one instance, one implementation of method step 800 of Figure 8 is carried out by capturing wide-angle portion 402 of IHADSS test pattern 400 with wide-angle camera 308, and thereafter recalling a representation of what wide-angle portion 402 of IHADSS test pattern 400 should look like if the IHADSS system were functioning substantially optimally. The underlying methodology of method step 800 is based on an agreed upon luminance difference which is expressed as square root of two

differences in gray or luminance levels (this standard is a de facto standard which has arisen over the years by course of use). Thus, in one embodiment, the captured image is iteratively moved through and a count is kept as to how many square root of two gray scale "jumps," or discontinuities (which occur in test pattern 400 at the boundaries of the varying gray bars – see Figure 4), are detected. This number of discontinuities are then compared against what is expected if the IHADSS is operating correctly. Specific examples of the foregoing appear in Specific Implementation A: Design of Interface and Algorithms for an Image Quality Tester (e.g., Figure A15 and Algorithm Design section, subsection D, which discusses identifying the number of gray shades within a test pattern), and Specific Implementation B: Preliminary Design of An Image Quality Tester for Helmet-Mounted displays.

In one instance, one implementation of method step 802 of Figure 8 is carried out by capturing wide-angle portion 402 of IHADSS test pattern 400 with wide-angle camera 308, and thereafter recalling a representation of what wide-angle portion 402 of IHADSS test pattern 400 should look like if the IHADSS system were functioning substantially optimally. The underlying methodology of method step 802 is based on a recognition that when the IHADSS is functioning properly, and the pilot has not improperly adjusted display size, the boundaries of the test pattern should be just on the edge of the field of view of the display. Accordingly, if the boundaries are found at the wrong location, or if the boundaries are not detected at all, it is known that there is a malfunction of some sort. Specific examples of the foregoing appear in Specific Implementation A: Design of Interface and Algorithms for an Image Quality Tester (see e.g., Figure A15 and Algorithm Design section, subsection E, which discusses identifying boundary lines), and Specific Implementation B, Preliminary Design of An Image Quality Tester for Helmet-Mounted displays.

In one instance, one implementation of method step 804 of Figure 8 is carried out by capturing wide-angle portion 402 of IHADSS test pattern 400 with wide-angle camera 308, and thereafter recalling a representation of what wide-angle portion 402 of IHADSS test pattern 400 should look like if the IHADSS system were functioning substantially optimally. The underlying methodology is based on a recognition that when the IHADSS is functioning properly, and the pilot has brightness and contrast levels of the IHADSS set within acceptable tolerances, the average gray level detected

should be within certain defined tolerances relative to average gray level drawn on how the image should appear if the IHADSS is functioning in a substantially optimal mode. Accordingly, if the detected gray levels are substantially outside the bounds of the defined tolerances, it is known either that the system is malfunctioning or that the pilot has either the brightness and/or contrast settings wrong. Specific examples of the foregoing appear in Specific Implementation A: Design of Interface and Algorithms for an Image Quality Tester (e.g., Figure A15 and Algorithm Design section, subsection E, which discusses identifying the contrast, brightness, and gray level relationship), and Specific Implementation B: Preliminary Design of An Image Quality Tester for Helmet-Mounted displays.

With reference now to Figure 9, depicted are alternate implementations of the high-level logic flowchart depicted in Figure 8. Method step 900 illustrates that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one gray-shades-displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 800). Method step 902 shows that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 802). Method step 904 shows that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one image quality figure of merit indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 804). The remaining method steps of the flowchart depicted in Figure 9 function substantially as shown and described herein.

Those skilled in the art will recognize that the state of the art has progressed to the point where there is little distinction left between hardware and software implementations of aspects of systems; the use of hardware or software is generally

(but not always, in that in certain contexts the choice between hardware and software can become significant) a design choice representing cost versus efficiency tradeoffs. The foregoing detailed description has set forth various embodiments of the devices and/or processes via the use of block diagrams, flowcharts, and examples. Insofar as such block diagrams, flowcharts, and examples contain one or more functions and/or operations, it will be understood as notorious by those within the art that each function and/or operation within such block diagrams, flowcharts, or examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or any combination thereof. In one embodiment, the present invention may be implemented via Application Specific Integrated Circuits (ASICs). However, those skilled in the art will recognize that the embodiments disclosed herein, in whole or in part, can be equivalently implemented in standard Integrated Circuits, as a computer program running on a computer, as a program running on a processor, as firmware, or as virtually any combination thereof and that designing the circuitry and/or writing the code for the software or firmware would be well within the skill of one of ordinary skill in the art in light of this disclosure. In addition, those skilled in the art will appreciate that the mechanisms of the present invention are capable of being distributed as a program product in a variety of forms, and that an illustrative embodiment of the present invention applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of a signal bearing media include but are not limited to the following: recordable type media such as floppy disks, hard disk drives, CD ROMs, digital tape, and transmission type media such as digital and analogue communication links using TDM or IP based communication links (e.g., packet links).

In a general sense, those skilled in the art will recognize that the various embodiments described herein which can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or any combination thereof can be viewed as being composed of various types of "electrical circuitry." Consequently, as used herein "electrical circuitry" includes but is not limited to electrical circuitry having at least one discrete electrical circuit, electrical circuitry having at least one integrated circuit, electrical circuitry having at least one application specific integrated circuit, electrical circuitry forming a general purpose computing device configurable by a computer program (e.g., a general purpose

computer configurable by a computer program or a microprocessor configurable by a computer program), electrical circuitry forming a memory device (e.g., any and all forms of random access memory), and electrical circuitry forming a communications device (e.g., a modem, communications switch, or optical-electrical equipment).

Those skilled in the art will recognize that it is common within the art to describe devices and/or processes in the fashion set forth above, and thereafter use standard engineering practices to integrate such described devices and/or processes into data processing systems. That is, the devices and/or processes described above can be integrated into data processing system via a reasonable amount of experimentation. Figures 7 and 8 show an example representation of a data processing system into which the described devices and/or processes may be implemented with a reasonable amount of experimentation.

With reference now to Figure 10, depicted a pictorial representation of a conventional data processing system in which illustrative embodiments of the devices and/or processes described herein may be implemented. It should be noted that a graphical user interface systems (e.g., Microsoft Windows 98 or Microsoft Windows NT operating systems) and methods can be utilized with the data processing system depicted in Figure 10. Data processing system 1020 is depicted which includes system unit housing 1022, video display device 1024, keyboard 1026, mouse 1028, and microphone 1048. Data processing system 1020 may be implemented utilizing any suitable computer such as a DELL portable computer system, a product of Dell Computer Corporation, located in Round Rock, Texas; Dell is a trademark of Dell Computer Corporation.

EXAMPLE IMPLEMENTATION A

(The following is similar to Hsieh, et al., "Design of Interface and Algorithms for an Image Quality Tester," USAARL Report No. 2000-26 (August, 2000), the content of which is hereby incorporated by reference in its entirety.)

USAARL REPORT NO. 2000-26

Design of Interface and Algorithms for an Image Quality Tester

By

Sheng-Jen Hsieh

Texas A&M University

and

Clarence E. Rash
John C. Mora

Aircrew Health and Performance Division

and

Thomas H. Harding

Howard H. Beasley
John S. Martin

UES, Inc.

August, 2000

U.S. Army Aeromedical Research Laboratory
Fort Rucker, Alabama 36362-0577

Acknowledgments

This work was supported by the US Army Research Office Scientific Services Program administered by Battelle (Delivery Order 549, Contract No. DAAL03-86-D-0001).

Table of contents

Page

Introduction 22

Functionality and operating process 22

Communication interface design 23

Data acquisition card configuration 24

POWER SPECIFICATIONS OF THE DAQ CARD 25

Sensors and DAQ card integration 26

Integration of sensor status information into software design 26

Design of image capture interface module and other features 27

Algorithm design 28

Conclusion and future directions 31

References 33

Appendix A. List of manufacturers 34

Appendix B. Program code 35

List of figures

A1. The IHADSS 22

A2. The IHADSS HDU 22

A3. Display size 23

A4. IHADSS built-in test pattern 23

A5. Previous design 23

A6. Revised design 23

A7. Location of sensors on proposed HMD fixture design 24

Table of contents (continued)

List of figures (continued)

Page

A8. Typical DAQCard-DIO-24 configuration 24

A9. Schematic diagram of proposed design 26

A10. Initial display screen, switches open 27

A11. Display screen, Continue button pressed, switches open 27

A12. Display screen, switches 1 and 2 pressed 27

A13. Display screen, image capture module activated 27

A14. Screenshot of image capture module 27

A15. Screenshots of parameter setting display screens 27

A16. Original image 29

A17. Image after binary processing 29

List of tables

1. Pin assignments for the CB terminal block 25

2. Power specifications for input and output signals 26

Introduction

The U.S. Army's AH-64 Apache attack helicopter incorporates a monocular helmet mounted display (HMD) known as the Integrated Helmet and Display Sighting System (IHADSS). The IHADSS consists of various electronic components and a helmet/display system called the Integrated Helmet Unit (IHU). The IHU (Figure A1) includes a helmet, visor housings with visors, miniature cathode ray tube (CRT), and helmet display unit (HDU). The HDU (Figure A2) serves as an optical relay device which conveys the image formed on the CRT through a series of lenses, off a beamsplitter (called a combiner), and into the aviator's right eye. The CRT is 1 inch in diameter and uses a P-43 phosphor. The combiner is a multilayer dichroic filter which is maximized for reflectance at the peak emission of the P-43 phosphor.

See
Figure A1. The IHADSS.

See
Figure A2. The IHADSS HDU.

Currently, there is no existing objective system for validation in the field of the quality of the imagery presented via the IHADSS. To maintain system integrity and readiness, and to provide pilots with validated pilotage, navigation, and fire control imagery, there is a need to develop an image quality testing tool for the IHADSS. This tester could be used as a validation tool to verify settings for regular flight missions and for preventive maintenance tasks. A preliminary tester design for the AH-64's IHADSS HMD was proposed and reported in Hsieh et al. (1999).

The objectives of the work reported here were to design and integrate communications interface and software procedure components for the proposed IHADSS HMD image tester. This is a continuation of the previous effort. The function of the communications interface is to sense (i.e., calculate) the positions of the camera and HMD based on the status of limit switches attached to a fixture used to mount the camera and HMD. This information then is converted to an eight-bit binary value using a digital I/O (input/output) data acquisition card. This discrete value is used by a custom developed software program as an indicator of the status of the hardware. Image capture routines then are activated to capture the test pattern generated by the HMD under test. The program applies image processing procedures to the images. In addition, image processing algorithms that can extract image features from HMD imagery and analyze them relative to the design specifications are proposed. These developments will allow automated evaluation of the image quality of an HMD.

FUNCTIONALITY AND OPERATING PROCESS

The IHADSS HMD has a 30-degree vertical by 40-degree horizontal field of view (FOV). Corner obscuration is permissible and symmetrical, as illustrated in Figure A3. The built-in test pattern (Figure A4) of the IHADSS HMD is used as the inspection specification on which the tester will be based. The test pattern shows strips of gray opposed along a centering line. Each strip contains a minimum of 8 to 10 shades of gray, depending on the contrast ratio. Adjacent shades have a square

root of 2 differential of luminance. For a more detailed discussion of the HMD test pattern features, see the Honeywell, Inc., study guide (1985) and Harding et al. (1995). For testing this test pattern, the inspection features of the image quality tester prototype include: (1) four center lines, (2) one horizontal line, (3) 8 to 10 gray shades, (4) boundary lines, and (5) illumination and focus.

See
Figure A3. Display size.

See
Figure A4. IHADSS built-in test pattern.

The operation procedures of the proposed HMD tester are as follows:

(1) The pilot adjusts the HMD settings;
(2) The crew chief inserts the HMD into a holding fixture;
(3) The system examines the center and horizontal line features of the test pattern using a narrow-angle lens camera;
(4) The system inspects the test pattern for image displacement and/or disorientation;
(5) The system examines the number of gray-shades, the focus, and boundary lines, using a 40 X 30 angle lens; and
(6) The system generates a final report.

Communication interface design

In previous work (Hsieh et al., 1999), a configuration in which two cameras face the HMD from different directions was proposed. This design approach is shown in Figure A5. However, due to a change in cameras, specifically in the size of the proposed cameras, this approach was deemed no longer feasible; therefore, a new approach with two cameras placed in a line and an HMD on a moveable rack was proposed. Figure A6 shows the revised design.

See
Figure A5. Previous design.

See
Figure A6. Revised design.

Before designing the communications interface for the HMD, it was important to identify in detail the operating procedure for the proposed HMD tester. This allowed determination of the number of sensors needed and the way the sensors would be integrated with the hardware. Below is a more detailed description of the sub-steps for steps (3) and (4) described above in section 2:

- Place the HMD in the fixture.
- Sensor #1 senses the HMD is present.
- Sensor #2 senses the HMD is facing camera #1.
- System captures the image.
- Crew chief moves the HMD to face camera #2.

- Sensor #3 senses the HMD in position #2.
- System captures the image.

Three sensors are required to accomplish the above sequence of events. One would be used to indicate the presence of the HMD, a second to indicate that the HMD is facing camera #1, and a third to indicate that the HMD is facing camera #2. Figure A7 shows the position of the sensors with respect to the HMD fixture.

See
Figure A7. Locations of sensors in proposed HMD fixture design.

DATA ACQUISITION CARD CONFIGURATION

A data acquisition card (DAQCard-DIO-24) by National Instruments was chosen to interface between the hardware sensors and the software. This card can fit into either of the notebook computer's PCMCIA slots.

In addition, an input/output (I/O) cable and terminal block are available to facilitate connecting the DAQ card to external devices such as panel meters, instruments, and solid-state relays. Figure A8 displays this configuration. Since the proposed tester is driven (in this prototype stage) by a notebook computer to minimize the size of the tester, the ability to use the PCMCIA slot as the I/O interface channel between the sensor hardware and system software was an essential feature.

See
Figure A8. Typical DAQCard-DIO-24 configuration.

This DAQCard-DIO-24 provides three configurable ports with 24 available digital I/O lines, which allow it to switch external devices such as transistors and solid-state relays, read the status of external device digital logic, and generate interrupts. Table 1 describes the cable pin assignment to the terminal block. Even numbers are signal grounds and odd numbers are the I/O signal lines. There are eight signals lines associated with each port (e.g., ports A, B, and C). Thus, the eight signal lines associated with port A are denoted as PA0, PA1, and so on, up to PA7. The same notation applies for ports B and C.

Table 1.
Pin assignments for the CB-50 terminal block.

GND	2	1	PC7
GND	4	3	PC6
GND	6	5	PC5
GND	8	7	PC4
GND	10	9	PC3
GND	12	11	PC2
GND	14	13	PC1
GND	16	15	PC0
GND	18	17	PB7
GND	20	19	PB6
GND	22	21	PB5
GND	24	23	PB4
GND	26	25	PB3
GND	28	27	PB2
GND	30	29	PB1
GND	32	31	PB0
GND	34	33	PA7
GND	36	35	PA6
GND	38	37	PA5
GND	40	39	PA4
GND	42	41	PA3
GND	44	43	PA2
GND	46	45	PA1
GND	48	47	PA0
GND	50	49	+5V

POWER SPECIFICATIONS OF THE DAQ CARD

As shown in Table 1, pin 49 provides +5 volts (V) from the PC Card I/O channel power supply. This pin is referenced to ground and can be used to power external digital circuitry that draws up to 1.0 amps. Note that there is a resettable thermal fuse that opens at voltages exceeding 1.0 amps and returns to normal operating conditions when cooled. The actual current available from this signal may be less than 200 milliamps depending on the computer. Table 2 describes the power specifications for input and put signals.

Table 2.
Power specifications for input and output signals.

Input signals			
	Level	Min	Max
	Input logic high voltage	2.2 V	5.3 V
	Input logic low voltage	-0.3 V	0.8 V
	Input current (0 < Vin < +5 V)	-1.0 mA (milliampere)	1.0 mA (milliampere)
Output signals			
	Pin 49 (at +5 V)	--	1.0 A

By default, all digital lines are pulled up to a logical HIGH setting. To keep a digital line in a logical LOW position, a 4.7 kΩ resistor from the digital line to ground can be connected in parallel with the external device. For example, to pull PC7 down to logical LOW, if the DAQCard-DIO-24 is connected to a CB-50 I/O terminal block (see Figure A7), pin 1 can be connected to any even numbered ground pin on the CB-50 pin I/O connector with a 4.7 kΩ resistor in between.

SENSORS AND DAQ CARD INTEGRATION

As described earlier, three sensors (i.e., limited switches) are used to sense the HMD position and presence. Figure A9 shows how the HMD hardware fixtures, sensors, I/O cable, and DAQCard-DIO-24 card are integrated. A pull down 4.7 kΩ resistor is utilized for each input signal pin. Input signal pins PA0, PA1, and PA2 are each connected to a limited switch. Reading the return value from the 8-bit I/O signals allows determination of which switch has been pressed. For instance, by default, the return value of an I/O signal is 255, since all the input pins are in logical HIGH position. A return value of 254 indicates that switch 1 has been pressed. If a limited switch is mounted close to camera #1, one can further interpret that HMD is facing camera 1. Figure A9 shows a schematic diagram of the proposed design.

See
Figure A9. Schematic diagram of proposed design.

INTEGRATION OF SENSOR STATUS INFORMATION INTO SOFTWARE DESIGN

The DAQCard-DIO-24 card is used to capture the sensors' status so that the software system can fuse this information with other sequences of events. For instance, knowing the status of limited switch #1 (which is mounted on the bottom of the enclosure) allows the system to determine if the HMD is present or not; and thus whether or not to activate the image capture routines. A Visual Basic function has been designed to query the hardware register that records the sensor status. Figures A10-A13 demonstrate the integration of an HMD setup and image capture modules

using feedback from the designed function. Switch #1, which indicates the presence of an HMD, will be mounted on the bottom of the enclosure. Switch #2, which indicates whether or not the HMD is facing the narrow-angle camera, will be mounted near the stopper on the rack by camera #1. Switch #3, which indicates whether or not the HMD is facing the wide-angle camera, will be installed near the stopper on the opposite side of the rack by camera #2.

See
Figure A10. Initial display, screen, switches open.

See
Figure A11. Display screen, Continue button pressed, switches open.

For details about the specifications and configuration of the DAQCard-DIO-24 card, please refer to the (1) DAQCard-DIO-24 user manual and (2) DAQ user manual for PC compatibles. For details about the code developed for this module, see Appendix B.

See
Figure A12. Display screen, switches 1 and 2 pressed.

See
Figure A13. Display screen, image capture module activated.

Design of image capture interface module and other features

Revisions to previously developed modules and additional features include the following:

- Image capture interface module: In the previous effort (Hsieh et al., 1999), object-linked embedding (OLE) techniques were used to launch the image capture driver included with the MRT Video-Port Professional software package. Based on Army recommendations, this module has been replaced with a new module written using the software's built-in tool-box library. Only the most essential functions are provided by this revised module. The revised module also provides an image format with a 780 x 510 pixel resolution. Figure A14 is a screenshot of the newly designed image capture interface module, showing an image of the HMD test pattern taken using the wide-angle camera.

See
Figure A14. Screenshot of image capture module.

See
Figure A15. Screenshots of parameter setting display screens.

- Password protection for system accessibility: A password is needed to enter the system or change parameter settings.

- Parameter setting features: Some parameters are camera dependent and/or user dependent. A password is needed to change parameter settings. For example, the size of a test pattern is camera lens angle dependent; thus, the gray-shade stripes'

height and width are proportional to the camera lens angle. In addition, some advanced system features should be user-restricted and available only to engineers. Figure A15 displays a parameter setting screen in which a test pattern is used as the background and the text boxes are displayed adjacent to the test pattern. This screen allows engineers to enter parameter values based on camera measurements. In addition, the image capture functions can be enabled or disabled.

ALGORITHM DESIGN

Following is a detailed description of the procedures used to evaluate key features of a test pattern such as center lines, center point, focus, test of resolution, and test pattern boundary. Two cameras with different angles are utilized to inspect different features within a test pattern. For instance, center line, center point and focus features are evaluated using the narrow-angle camera. On the other hand, features such as test pattern contrast and boundary characteristics are evaluated by using the wide-angle camera.

These procedures detail the steps followed by the algorithm for each feature. The information is compiled based on the available data, which were taken from ***three*** different HMD units. In designing the algorithms, the following issues were taken into consideration.

- <u>Data collection</u>: Images of the test pattern as taken by a narrow-angle and a wide-angle camera were captured for the purposes of designing the specifications, creating possible noise, and testing the proposed algorithms. These included images taken from different orientations (e.g., +/- 5 degrees of rotation), different displacements, in/out of focus, and varying contrast/brightness ratios.

- <u>Specification design</u>: Correlation coefficients were frequently computed to identify the relationships between variables such as the image focus magnitude and gray scale variation. Strong positive or negative correlations between variables allow the use of one variable to predict another. For instance, there appears to be a strong negative relationship between image focus magnitude and gray scale variation. In other words, by knowing the gray scale variation, we can predict whether the HMD is in focus or not. Moreover, with sufficient data, it is possible to predict the extent of the lack of focus.

- <u>Designed noise</u>: Knowing the types of noise present in the data helps the tester to differentiate between good and bad images. Although limited data were available to allow this, a few anticipated sources of noise were created to simulate real ones, and used to verify the proposed algorithms. Primary designed noises were displacement, orientation, and focus.

Algorithms were developed to detect various features within the test pattern as described earlier. Some of the basic ideas were proposed in previous work (Hsieh et al., 1999). Modifications were made due to the availability of the camera. (Previously, images were created using graphics software. These images were of

course different from actual images captured from the proposed camera.) These procedures are described below according to the feature of interest.

A. Identify the number of center lines.

Step 1. Apply binary image technique to the entire image.
Step 2. Draw multiple lines across X and/or Y axes.
Step 3. Identify mask with feature of B/W...W/B.
Step 4. Store the intersection points in an array with multiple dimensions.
Step 5. Construct regression lines based on the points within each dimension.
Step 6. Develop regression lines to compare the parallel property.
Step 7. Average the intersection points around the array to obtain the number of estimated lines,
where B = black pixel and W = white pixel.

Note: Use of linear regression analysis would make the linear mode robust and insensitive to noise presence.

How to find the threshold value needed to conduct the binary image process:

Step 1. Capture an image P(m, n) with m = 0,1,2....,M and n = 0,1,2...,N.
Step 2. Calculate the center/horizontal lines in area A.
Step 3. Compute the ratio $\gamma = A/P(m, n)$.
Step 4. Find α knowing that the probability $p(x >= \mu + \alpha s) = \gamma/2$.
Step 5. Construct binary image knowing that the threshold value $T= \mu + \alpha s$.

Where u is the mean and S is the standard deviation of the gray level of the image, and γ represents the percentage of the center four-line region relative to the overall image area. The center four lines are the ones that have a higher gray level than the rest of the background; $\gamma/2$ will provide a better contrast of the center four-line area.

For example: Given an image P(m, n) as shown in Figure A16.

Step 1. P(m, n) where m = 0, ...,780 and n = 0,...510
Step 2. Center area A is approximate to H+V-O
H: horizontal line, V: four vertical lines, O: center overlap region
H = 54 X 485, V = 758 X 10, and O = 54 X 12
A = 33122
Step 3. $\gamma = A/P(m, n) = 33122/(780*510) = 0.0832$
Step 4. $p(x >= \mu + \alpha s) = 0.0416$, where $\mu = 24.14$, $s = 29.67$; therefore, $\alpha = 2.652$
Step 5. $T= \mu + \alpha s$; therefore, T = 102.82

Figure A17 shows the image after binary processing.

See
Figure A16. Original image.

See
Figure A17. Image after binary processing.

B. Identify the center point.

Approach #1:

Step 1. Construct a regression line based on all the intercepted points. By doing so, a black line perpendicular to the horizontal line will be formed.

Step 2. Identify the mid-point of an array as a starting point with the feature of W/B...B/W.

Step 3. Examine neighboring pixels to see if a W/W/W mask exists.

Step 4. If a W/W/W mask exists, stop the procedure; else next step.

Step 5. Check the distance of neighboring pixels from the regression line using a 3 x 3 area.

Step 6. Select the point with the smallest distance from the regression line as the next point.

Step 7. Go to step 3.

Approach #2:

Step 1. Calculate the center region of the test pattern as area A.

Step 2. Arrange the pixel gray level in decreasing order.

Step 3. Select the first A number of pixels.

Step 4. Find the p(x, y) with the lowest gray level within the A number of pixels.

Step 5. Compute the binary image based on the threshold value of p(x, y).

Step 6. Calculate the center of mass:

$$\text{Center\_X} = \Sigma X_i / A; \ \text{Center\_Y} = \Sigma Y_i / A$$

Note: Approach #2 is good only under the assumption that there are no noises that have the same gray level as the pixels within region A.

C. Identify test pattern orientation and displacement.

Step 1. Given an image P(m,n) with m=0,1,2...,M and n=0,1,2,...,N;

Step 2. Compute a theoretical center as point C, where C=(M/2, N/2);

Step 3. Identify the actual center point (based on part B, approach #1) as point B;

Step 4. Connect point C and point B to form the segment S_d;

Step 5. Compute the distance between points C and B as d, where d is the displacement;

Step 6. If segment S_d is parallel to the theoretical horizontal line or if the theoretical four-center lines are parallel with the actual four-center lines, then the orientation angle is 0; stop the process. Else go to next step;

Step 7. Form a lines segment S_a across point B, parallel to the theoretical four-center line, and intercepting the theoretical horizontal line; this intercept point is called point H. Form another line segment S_h to connect point C and point H.

Step 8. The angle between the line segments S_d and S_h is the orientation angle.

D. Identify the number of gray shades within a test pattern.

Approach #1:

Step 1. Use the center point as a starting point.

Step 2. Pick five points across the four vertical lines that are within the boundary of the gray shades.

Step 3. Compute the average gray level of the five points.

Step 4. Store it in one dimension of the array.
Step 5. If the boundary is not reached, move up or down a given distance, and go to Step 3. Else, go to next step.
Step 6. Use the square root of 2 to determine the number of gray shades.

Approach #2:
Step 1. Identify $g(x, y)_h$ and $g(x, y)_l$
Step 2. Compute the ratio $\gamma = g(x, y)_h / g(x, y)_l$
Step 3. Repeat the same process for the four vertical lines and gray shade regions.
Where $g(x, y)_h$ represents the pixel $p(x, y)$ with the highest gray level, and $g(x, y)_l$ represents the pixel $p(x, y)$ with the lowest gray level.

E. Identify boundary lines.

Step 1. Use the center point and boundary ratio to determine the region of the image boundary.
Step 2. Locate a starting point white pixel to use for backtracking through the rest of the white pixels for each line segment.

F. Identify the focus setting.

Step 1. Use the line scan technique to record the pixels along the four vertical lines.
Step 2. Use the B/W/B mask to identify the separation of lines.
Step 3. Compute the ratio of bottom to mid-peak and peak to valley for all four lines.
Step 4. If the ratio is approximately one, conclude that the focus setting is good; or else check the focus setting.

G. IDENTIFY THE CONTRAST, BRIGHTNESS AND GRAY LEVEL RELATIONSHIP.

Step 1. User enters the current brightness and contrast.
Step 2. System computes the average image gray level.
Step 3. System calculates the corresponding gray level variance based on a derived function.
Step 4. System computes the predicted focus magnitude.

Conclusion and future directions

In this project, an interface was designed to allow communication between the sensors and the software application. This interface consists of designed circuitry, a data acquisition card, and an I/O connector. It fits into a standard PCMCIA slot in a notebook computer. A fixture design that incorporates in-line cameras with an HMD holder is proposed for image capture. In addition, a new image capture software application was developed utilizing the tool library included in the MRT Video-Port

Professional image grabber software package. Algorithms were designed, taking into consideration the steps of data collection, design specifications, and noise generation. Three HMD units were utilized to capture image data. Images with noise such as displacement, orientation, and focus were captured. Statistical approaches such as correlation coefficients and regression analysis were utilized to probe the relationship between performance/image variables (such as focus) and image gray level variation. Knowledge of these relationships allows use of image variables to verify and/or predict control variables such as focus resolution.

Image measurement specifications were developed based on statistical analysis of the collected image data. Algorithms for detecting four vertical lines, center point, focus, and boundary are proposed. Examples are given to illustrate how the procedures work and screenshots of the before and after image processing are shown.

Future work will likely include:

- Coding of the designed image specification and algorithms and verification with image data collected from the field.
- Fabrication of the image tester with a robust fixture holder which has three spring loaded jags to provide constant pressure around the HMD and to accommodate variation between HMDs.
- Field evaluation of tester accuracy.

References

Harding, T.H., Beasley, H.H., Martin, J.S. and Rash, C.E. 1995. Physical Evaluation of the Integrated Helmet and Display Sighting System Helmet Display Unit. Fort Rucker. AL: U.S. Army Aeromedical Research Laboratory. USAARL Report No. 95-32.

Avionics Systems Group, Military Avionics Division. 1985. Integrated Helmet and Display Sighting System - Study Guide. St. Louis Park, MN: Honeywell, Inc.

Hsieh, S., Rash, C.E., Harding, T.H., Beasley, H.H., and Martin, J.S. 1999. Preliminary Design of an Image Quality Tester for Helmet-Mounted Displays. Fort Rucker, AL: U.S. Army Aeromedical Research Laboratory. USAARL Report No. 2000-08.

APPENDIX A.

List of manufacturers.

National Instruments
11500 Mopac Expressway
Austin, TX 78759-3504

APPENDIX B.

Program code.

EXIT SETTING FORM

```
Option Explicit

Private Sub Image2_Click()

End Sub

Private Sub nolButton_Click()

password.Show
Unload exitwnd

End Sub
Private Sub yesButton_Click()

Unload password
Unload Me

End Sub
```

820F1 FORM

```
Private Sub Frame1_DragDrop(Source As Control, X As Single, Y As Single)

End Sub

Private Sub Timer1_Timer()

Dim PauseTime, Start

  PauseTime = 2   ' Set duration.
  Start = Timer   ' Set start time.
  Do While Timer < Start + PauseTime
    DoEvents    ' Yield to other processes.
  Loop
    Unload Me
  password.Show

End Sub
```

DIONEPORTBACKUP FORM

```
Option Explicit
Option Base 0
'
' Constant for PrintText
'
Const LEN_PRINTTEXT = 4096
```

```
'***************************************************************************
'SUBROUTINE:  PrintText
'DESCRIPTION: PrintText to desired TextBox (upto 4096 characters)
'INPUTS:     txtBox - TextBox to print on
'            strText - Text to print
'***************************************************************************
Sub PrintText(txtBox As TextBox, strText As String)

  txtBox.Text = Right$(txtBox.Text + strText$ + Chr$(13) + Chr$(10), LEN_PRINTTEXT)

  txtBox.SelStart = Len(CStr(txtBox.Text))

  DoEvents

End Sub

'***************************************************************************
'SUBROUTINE:  cmdExit_Click
'DESCRIPTION: Clean up and exit
'***************************************************************************
Sub cmdExit_Click()

  End

End Sub

'***************************************************************************
'SUBROUTINE:  Form_Load
'DESCRIPTION: Gets automatically called at startup
'***************************************************************************
Sub Form_Load()

If (FlagLabel = 0) Then
   Step1.BackColor = &H80FF80
   Call PrintText(txtStatusBox, "Place HMD in camera #1 position ! ")
 Else
   Step1.BackColor = &HFFFFFF    'White
   Step1.ForeColor = &H8000000F  'Black
End If

End Sub

'***************************************************************************
'SUBROUTINE:  cmdContinue_Click
'DESCRIPTION: The main NI-DAQ operations are here
'***************************************************************************
Sub cmdContinue_Click()

  '
  ' Local Variable Declarations:

  Dim iStatus As Integer
  Dim iRetVal As Integer
  Dim iDevice As Integer
  Dim iPort As Integer
  Dim iMode As Integer
  Dim iDir As Integer
  Dim iPattern As Long
  Dim iIgnoreWarning As Integer
  Dim PauseTime, Start

  iDevice% = 1

  'Temporarily disable buttons for protection from 'DoEvents'
  cmdContinue.Enabled = False
```

```
    cmdExit.Enabled = False

  '  Configure port as input, no handshaking.

    iStatus% = DIG_Prt_Config(iDevice%, iPort%, iMode%, iDir%)
    iRetVal% = NIDAQErrorHandler(iStatus%, "DIG_Prt_Config", iIgnoreWarning%)
    iStatus% = DIG_In_Prt(iDevice%, iPort%, iPattern&)
    iRetVal% = NIDAQErrorHandler(iStatus%, "DIG_In_Prt", iIgnoreWarning%)

'************************************************************
' PA0 -- HMD position switch;      On/254, Off/255
' PA1 -- HMD in camera #1 position; On/253, Off/255
' PA2 -- HMD in camera #2 position; On/251, Off/255
' HMD lucked in camera #1; then, iPattern& = 252
' HMD lucked in camera #2; then, iPattern& = 250
' Cover is opened; then, iPattern& = 255
'************************************************************

    If ((iPattern& = 251) Or (iPattern& = 253) Or (iPattern& = 255)) Then

     Call PrintText(txtStatusBox, "Enclosure is open !")
    End If

    If ((iPattern& = 252) And (FlagLabel = 0)) Then

     Call PrintText(txtStatusBox, "Positon is lucked ! Loading program ")
     Step1.BackColor = &HFFFFFF    'White
     Step1.ForeColor = &H8000000F  'Black
     Step2.BackColor = &H80FF80    'Light Green


       PauseTime = 2   ' Set duration.
       Start = Timer   ' Set start time.
       Do While Timer < Start + PauseTime
         DoEvents    ' Yield to other processes.
       Loop

     Unload Me 'Form1Backup
     Image_Capture.Show

     Step1.BackColor = &HFFFFFF    'White
     Step1.ForeColor = &H8000000F  'Black
     Step2.BackColor = &HFFFFFF    'White
     Step2.ForeColor = &H8000000F  'Black
     Step3.BackColor = &H80FF80    'Light Green

     Call PrintText(txtStatusBox, "Place HMD in position indicated for use with wide-angle camera; then press
  Continue button below")

     FlagLabel = FlagLabel + 1    'Flag for sequencing the events


    End If

    If ((iPattern& = 250) And (FlagLabel = 1)) Then

     Call PrintText(txtStatusBox, "Positon is lucked ! Loading program ")

     Step1.BackColor = &HFFFFFF    'White
     Step1.ForeColor = &H8000000F  'Black
     Step3.BackColor = &HFFFFFF    'White
     Step3.ForeColor = &H8000000F  'Black
     Step4.BackColor = &H80FF80    'Light Green

     PauseTime = 2   ' Set duration.
```

```
        Start = Timer   ' Set start time.
        Do While Timer < Start + PauseTime
          DoEvents    ' Yield to other processes.
        Loop

       Unload Me
       Image_Capture.Show

       FlagLabel = FlagLabel + 1

     End If

     If (FlagLabel = 2) Then

       Step1.BackColor = &HFFFFFF    'White
       Step1.ForeColor = &H8000000F  'Black
       Step2.BackColor = &HFFFFFF    'White
       Step2.ForeColor = &H8000000F  'Black
       Step3.BackColor = &HFFFFFF    'White
       Step3.ForeColor = &H8000000F  'Black
       Step4.BackColor = &HFFFFFF    'White
       Step4.ForeColor = &H8000000F  'Black
       Step5.BackColor = &H80FF80   'Light Green

       Call PrintText(txtStatusBox, "Press Return button below to return to main menu ")
       FlagLabel = FlagLabel + 1

     End If

     If (FlagLabel < 3) Then
       cmdContinue.Enabled = True
     Else
       cmdContinue.Enabled = False
     End If

     cmdExit.Enabled = True

End Sub

Private Sub Image1_Click()

End Sub

Private Sub Return_Click()

  If (FlagLabel = 3) Then
    Step5.BackColor = &HFFFFFF    'White
    Step5.ForeColor = &H8000000F  'Black
  End If
  Unload Form1Backup
  Form2.Show

End Sub
```

820F2 FORM

```
Private Sub Form_Load()

If (DummyY = 0) Then
  cmdImage_Analysis.Enabled = False
  cmdResults.Enabled = False
  cmdSetup.Enabled = True

Else
```

```
cmdImage_Analysis.Enabled = True
cmdResults.Enabled = True
cmdSetup.Enabled = False
End If

End Sub
Private Sub cmdSettings_Click()

Unload Form2
passwordforsettings.Show
cmdSettings.Enabled = False 'once it is set; u can't go back

End Sub

Private Sub cmdSetup_Click()

cmdSetup.ToolTipText = "Set up the HMD"

Unload Form2
Form1Backup.Show
DummyY = 1  'to de-activate the functions

End Sub
Private Sub cmdImage_Capture_Click()

cmdImage_Capture.ToolTipText = "Image capture of the HMD"

Unload Form2

cmdImage_Capture.Enabled = False

Image_Capture.Show

End Sub

Private Sub cmdImage_Analysis_Click()

cmdImage_Analysis.ToolTipText = "Image features analysis"

Unload Form2
Form4.Show
MsgBox "Select an image file"

End Sub

Private Sub cmdResults_Click()

cmdResults.ToolTipText = "Analysis findings"

Unload Form2
Form5.Show

End Sub

Private Sub cmdQuit_Click()

cmdQUIT.ToolTipText = "Exit from the system"

Unload Form2
End

End Sub

Private Sub Image1_Click()
```

```
End Sub
```

820F479_00 FORM

```
Public Displacement, Angle As Double
Public CenterLineSlope As Double
Public CenterLineIntercept As Double
Public Center_Point_X, Center_Point_Y As Double

Const intUpperBoundX = 780 '320 total
Const intUpperBoundY = 510 '244 total
Const n = 4 '# of center line

Dim X, Y As Integer
Dim picObject0, picObject1 As Image 'Do not delete picObject1; U used picObject1 somewhere in the form
Dim picObject3 As Picture
Dim Coord_X(0 To 45, 0 To 10) As Integer
Dim Coord_Y(0 To 45, 0 To 10) As Integer
Dim pixels(0 To intUpperBoundX, 0 To intUpperBoundY) As Long
Dim ImagePixels(2, intUpperBoundX, intUpperBoundY) As Integer

Private Sub Back_Click()

cmdBack.ToolTipText = "Back to previous stage"
Unload Form4
Form2.Show

End Sub

Private Sub cmdFocus_Click()

cmdFocus.ToolTipText = "Focus Measurement"

'********************************************
'Step 1: Calculate the image standard deviation
'Step 2:
'********************************************

Set Picture0.Picture = picObject0
For X = 0 To intUpperBoundX - 1
  For Y = 0 To intUpperBoundY - 1
    Picture0.PSet (X, Y), Picture0.Point(X, Y) - 10
  Next Y
Next X


End Sub

Private Sub Form_Load()

cmdFocus.Enabled = False
cmdGray_Shade.Enabled = False
cmdCenter_and_Boundary.Enabled = False
cmdEdgeDetection.Enabled = False
cmdGray_Shade.Enabled = False
cmdDis_and_Orientation.Enabled = False

End Sub
Private Sub cmdGray_Shade_Click()

cmdGray_Shade.ToolTipText = "Detecting number of gray shades"

End Sub
```

```
Private Sub cmdSelect_Click()

Dim filename, EdgeDetection As String
Dim bytRed, bytGreen, bytBlue, bytAverage As Integer
Dim GrayLong As Long
Dim SumGrayLevel, MeanGray, SumSquare, StandardDeviation, ThresholdValue As Double

cmdSelect.ToolTipText = "Select an image file first"

On Error GoTo FileError
If (Right$(Dir1.Path, 1) = "\") Then
  filename = File1.Path & File1.filename
 Else
  filename = File1.Path & "\" & File1.filename
End If

Open filename For Input As #1

Set picObject0 = LoadPicture(filename)
Set Picture0.Picture = picObject0

Close #1

'Do not reverse the sequence: image1 and picture0


Open "c:\windows\desktop\ImageMap.txt" For Output As #2

For X = 0 To intUpperBoundX - 1
  For Y = 0 To intUpperBoundY - 1

    pixels(X, Y) = Picture0.Point(X, Y)

    If (pixels(X, Y) = 0) Then
     bytRed = 0
     bytGreen = 0
     bytBlue = 0
    End If

    If (pixels(X, Y) > 0) Then
     bytRed = GetRed(pixels(X, Y))
     bytGreen = GetGreen(pixels(X, Y))
     bytBlue = GetBlue(pixels(X, Y))
    End If

    If (Y = 150) Then
    bytAverage = (bytBlue + bytRed + bytGreen) / 3
    Write #2, X, Y, bytRed, bytBlue, bytGreen, bytAverage
    End If

    'ImagePixels(0, X, Y) = bytRed
    'ImagePixels(1, X, Y) = bytGreen
    'ImagePixels(2, X, Y) = bytBlue
    'the file u have is in gray scale; therefore, u do not need to average

    Picture0.PSet (X, Y), RGB(bytRed, bytGreen, bytBlue)

  Next Y
Next X

Close #2

'SumGrayLevel = 0
'For X = 0 To intUpperBoundX - 1
'  For Y = 0 To intUpperBoundY - 1
```

```
'    SumGrayLevel = SumGrayLevel + ImagePixels(0, X, Y)
'  Next Y
'Next X

'MeanGray = SumGrayLevel / (intUpperBoundX * intUpperBoundY)
'SumSquare = 0

'For X = 0 To intUpperBoundX - 1
'  For Y = 0 To intUpperBoundY - 1
'    SumSquare = SumSquare + ((ImagePixels(0, X, Y) - MeanGray) * (ImagePixels(0, X, Y) - MeanGray))
'  Next Y
'Next X

'StandardDeviation = SumSquare / ((intUpperBoundX * intUpperBoundY) - 1)
'ThresholdValue = MeanGray + (2.5 * StandardDeviation)


If File1.filename = "Narrow.bmp" Then
cmdEdgeDetection.Enabled = True
cmdDis_and_Orientation.Enabled = True
cmdFocus.Enabled = True
MsgBox "Select one of the criterion on the left top corner"
End If

If File1.filename = "Wide.bmp" Then
cmdGray_Shade.Enabled = True
cmdCenter_and_Boundary.Enabled = True
MsgBox "Select one of the criterion on the left bottom corner"
End If

Exit Sub

FileError: MsgBox "Select an image file first !"

End Sub

Private Sub cmdCenter_and_Boundary_Click()

cmdCenter_and_Boundary.ToolTipText = "Finding the center lines and boundary"

Set Picture0.Picture = picObject0
For X = 0 To intUpperBoundX - 1
  For Y = 0 To intUpperBoundY - 1
    Picture0.PSet (X, Y), Picture0.Point(X, Y)
  Next Y
Next X

Set picObject3 = Picture0.Picture
SavePicture picObject3, "TEST1.BMP"
LoadPicture ("TEST1.BMP")

MsgBox "Boundary rountine ended !"

End Sub

Private Sub cmdEdgeDetection_Click()

Dim RGBLong As Long
Dim G_X, G_Y, G_X_Y As Integer
Dim bRXY, bRXm1Y, byRXYm1, bRXm1Ym1 As Integer
Dim bRXp1Y, bRXYp1, bRXp1Yp1, bRXp1Ym1, bRXm1Yp1 As Integer
Dim bytRed, bytGreen, bytBlue As Integer

cmdEdgeDetection.ToolTipText = "Edge Detection"

Set Picture0.Picture = picObject0
```

```
For X = 0 To intUpperBoundX - 1
  For Y = 0 To intUpperBoundY - 1

   If (X = 0 Or X = intUpperBoundX - 1 Or Y = 0 Or Y = intUpperBoundY - 1) Then

     bytRed = ImagePixels(0, X, Y)
     bytBlue = ImagePixels(1, X, Y)
     bytGreen = ImagePixels(2, X, Y)
     RGBLong = RGB(bytRed, bytGreen, bytBlue)

     Picture0.PSet (X, Y), RGBLong

   Else

    G_X = 0
    G_Y = 0
    G_X_Y = 0

    bRXY = ImagePixels(0, X, Y)
    bRXYp1 = ImagePixels(0, X, Y + 1)
    bRXm1Y = ImagePixels(0, X - 1, Y)
    bRXYm1 = ImagePixels(0, X, Y - 1)
    bRXm1Yp1 = ImagePixels(0, X - 1, Y + 1)
    bRXm1Ym1 = ImagePixels(0, X - 1, Y - 1)
    bRXp1Y = ImagePixels(0, X + 1, Y)
    bRXp1Ym1 = ImagePixels(0, X + 1, Y - 1)
    bRXp1Yp1 = ImagePixels(0, X + 1, Y + 1)

    G_X = bRXp1Ym1 + 2 * bRXp1Y + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXm1Y - bRXm1Yp1
    G_Y = bRXm1Yp1 + 2 * bRXYp1 + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXYm1 - bRXp1Ym1
    G_X_Y = Sqr((G_X * G_X) + (G_Y * G_Y))

    bytRed = G_X_Y

    bRXY = ImagePixels(1, X, Y)
    bRXYp1 = ImagePixels(1, X, Y + 1)
    bRXm1Y = ImagePixels(1, X - 1, Y)
    bRXYm1 = ImagePixels(1, X, Y - 1)
    bRXm1Yp1 = ImagePixels(1, X - 1, Y + 1)
    bRXm1Ym1 = ImagePixels(1, X - 1, Y - 1)
    bRXp1Y = ImagePixels(1, X + 1, Y)
    bRXp1Ym1 = ImagePixels(1, X + 1, Y - 1)
    bRXp1Yp1 = ImagePixels(1, X + 1, Y + 1)

    G_X = bRXp1Ym1 + 2 * bRXp1Y + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXm1Y - bRXm1Yp1
    G_Y = bRXm1Yp1 + 2 * bRXYp1 + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXYm1 - bRXp1Ym1
    G_X_Y = Sqr((G_X * G_X) + (G_Y * G_Y))

    bytBlue = G_X_Y

    bRXY = ImagePixels(2, X, Y)
    bRXYp1 = ImagePixels(2, X, Y + 1)
    bRXm1Y = ImagePixels(2, X - 1, Y)
    bRXYm1 = ImagePixels(2, X, Y - 1)
    bRXm1Yp1 = ImagePixels(2, X - 1, Y + 1)
    bRXm1Ym1 = ImagePixels(2, X - 1, Y - 1)
    bRXp1Y = ImagePixels(2, X + 1, Y)
    bRXp1Ym1 = ImagePixels(2, X + 1, Y - 1)
    bRXp1Yp1 = ImagePixels(2, X + 1, Y + 1)

    G_X = bRXp1Ym1 + 2 * bRXp1Y + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXm1Y - bRXm1Yp1
    G_Y = bRXm1Yp1 + 2 * bRXYp1 + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXYm1 - bRXp1Ym1
    G_X_Y = Sqr((G_X * G_X) + (G_Y * G_Y))

    bytGreen = G_X_Y
```

```
      Picture0.PSet (X, Y), RGB(bytRed, bytGreen, bytBlue)

    End If

  Next Y
Next X

End Sub
Private Sub cmdGray_Shade__Click()

Set Picture0.Picture = picObject0
For X = 0 To intUpperBoundX - 1
  For Y = 0 To intUpperBoundY - 1
     Picture0.PSet (X, Y), Picture0.Point(X, Y) - 5
  Next Y
Next X

MsgBox "Gray shades routine ended !"

End Sub

Private Sub cmdDis_and_Orientation_Click()

Const interval_range = 7

Dim WhitePixel, BlackPixel As Long
Dim linescan As Integer
Dim i, j, k, l, IntX, Temp_X, Temp_Y As Integer
Dim Flag, SumTline, Dummy As Integer
Dim interval As Integer
Dim ZeroO_X, ZeroO_Y As Double
Dim L1SlopeR, L2SlopeR, L3SlopeR, L4SlopeR, L1SlopeY, _
   L2SlopeY, L3SlopeY, L4SlopeY, AvgSlope As Double

Dim UpperBound, LowerBound As Double
Dim InterceptY As Integer
Dim Count_Points(0 To 403) As Integer
Dim TempInt, Choice As Integer
Dim Dum(0 To 15) As Double
Dim TempDouble As Double
Dim Tline(0 To 50) As Integer
Dim Oripixels(0 To intUpperBoundX, 0 To intUpperBoundY) As Long

Dim Displacement, Angle, Theta As Double
Dim CenterLineSlope As Double
Dim CenterLineIntercept As Double
Dim Center_Point_X, Center_Point_Y As Double
Dim TempText As String

cmdDis_and_Orientation.ToolTipText = "Displacement and Orientation"

Open "c:\windows\desktop\InspResults.txt" For Output As #1

For X = 0 To intUpperBoundX - 1
 For Y = 0 To intUpperBoundY - 1
  Oripixels(X, Y) = pixels(X, Y)
 Next Y
Next X

'Apply the binary image technique

For X = 0 To intUpperBoundX - 1
 For Y = 0 To intUpperBoundY - 1
  If (Oripixels(X, Y) < RGB(255, 255, 255)) Then
      Oripixels(X, Y) = 0
```

```
        Else
            Oripixels(X, Y) = RGB(255, 255, 255)
        End If
        Picture0.PSet (X, Y), Oripixels(X, Y)

    Next Y
  Next X

  'Find the number of center lines
  'A line is BW...WB; if there is less than four BW...WBs; then Image is tilled
  'white interval should be less than 7 for the central lines
  'use Black/White/Black to find a line

  linescan = 0
  interval = 1

  For Y = 50 To intUpperBoundY - 1
     Tline(linescan) = 0
     Flag = 0
     l = 0

     For X = 0 To intUpperBoundX - 1
        If ((Oripixels(X, Y) = RGB(0, 0, 0)) And _
        (Oripixels(X + 1, Y) = RGB(255, 255, 255))) Then

           For interval = 1 To interval_range - 1
             If (Oripixels(X + 1 + interval, Y) = RGB(0, 0, 0)) Then

                Tline(linescan) = Tline(linescan) + 1
                Flag = 1

                Coord_X(linescan, l) = X + 1           'of each line
                Coord_Y(linescan, l) = Y

                l = l + 1

             End If
             interval = interval_range 'stop the for loop
           Next interval
        End If
     Next X
     Y = Y + 10 '5          'to have 40 arbitary verticle lines
     If (Flag = 1) Then
     linescan = linescan + 1
     End If
  Next Y

  k = 0
  SumTline = 0
  For j = 0 To linescan - 1 'from prev. routine # of arb. ver. lines
   If (Tline(j) > 0) Then
     SumTline = SumTline + Tline(j)
     k = k + 1
   End If
  Next j

  If (3.5 <= (SumTline / k) <= 4.5) Then
    MsgBox ("Number of center lines is  " & n)

    L1SlopeR = GetSlope(linescan, 0, 0)
    L1SlopeY = GetSlope(linescan, 0, 1)

    L2SlopeR = GetSlope(linescan, 1, 0)
    L2SlopeY = GetSlope(linescan, 1, 1)

    L3SlopeR = GetSlope(linescan, 2, 0)
```

```
L3SlopeY = GetSlope(linescan, 2, 1)

L4SlopeR = GetSlope(linescan, 3, 0)
L4SlopeY = GetSlope(linescan, 3, 1)

AvgSlope = (L1SlopeY + L2SlopeY + L3SlopeY + L4SlopeY) / 4
LowerBound = 0.025 * AvgSlope
UpperBound = 1.025 * AvgSlope

'Use the absolute value; therefore, it works on both -/+ values

If ((Abs(LowerBound) <= Abs(L1SlopeY) <= Abs(UpperBound)) And _
   (Abs(LowerBound) <= Abs(L2SlopeY) <= Abs(UpperBound)) And _
   (Abs(LowerBound) <= Abs(L3SlopeY) <= Abs(UpperBound)) And _
   (Abs(LowerBound) <= Abs(L4SlopeY) <= Abs(UpperBound))) Then
   MsgBox ("Four lines are parallel !")
Else: MsgBox ("Potential errors in finding parallel lines")
End If

Else
  MsgBox ("Number of center lines is " & SumTline / k)
End If

'The following is to find the center point of the image
'Step 1: Find the black pixel
'Step 2: Calcuate the neighborhood pixels distance to the regression line
'Step 3: Locate the one that has the smallest distance
'Step 4: Check to see if the feature of w
'                         wwww
'                          w
'been meet
'if not; based on current X, Y; go to Step 1

BlackPixel = RGB(0, 0, 0)
WhitePixel = RGB(255, 255, 255)

CenterLineSlope = GetSlope(linescan, 0, 2)
CenterLineIntercept = GetSlope(linescan, 0, 3)

MsgBox ("C.L.Intercept = " & CenterLineIntercept)
MsgBox ("C.L.Slope = " & CenterLineSlope)

For Y = 20 To intUpperBoundY - 1
 X = (Y * CenterLineSlope) + CenterLineIntercept
  IntX = X
  If (Oripixels(IntX, Y) = BlackPixel) Then

    l = 0
    Temp_X = 0
    Temp_Y = 0
    For i = -1 To 1
      For j = -1 To 1
        If (Oripixels(IntX + i, Y + j) = WhitePixel) Then
          Temp_X = Temp_X + (IntX + i)
          Temp_Y = Temp_Y + (Y + j)
          l = l + 1
        End If
        If (l >= 3) Then      'Neighborhood pixels are White
          Center_Point_X = Temp_X / l
          Center_Point_Y = Temp_Y / l
          MsgBox ("Center X = " & Center_Point_X)
          Beep
          MsgBox ("Center Y = " & Center_Point_Y)
          i = 1
          j = 1
          Y = intUpperBoundY
```

```
          End If
        Next j
      Next i

      l = 0
      Dum(l) = 0
      For i = 0 To 1
        For j = 0 To 1
          If (i <> 0 Or j <> 0) Then
            Dum(l) = Measure_Distance(CenterLineIntercept, CenterLineSlope, X + i, Y + j)
            l = l + 1
          End If
        Next j
      Next i

      For k = 0 To l - 1
        If (Dum(k) < Dum(k + 1)) Then
          TempDouble = Dum(k)
          Dum(k) = Dum(k + 1)
          Dum(k + 1) = TempDouble
        End If
      Next k

      For i = 0 To 1
        For j = 0 To 1
          If ((i <> 0 Or j <> 0) And (Dum(l - 1) = Measure_Distance(CenterLineIntercept, CenterLineSlope, X + i,
Y + j))) _
            Then _

            X = X + i
            Y = Y + j - 1  'because Y auto. inc. by 1
            i = 1
            j = 1

          End If
        Next j
      Next i

    End If
  Picture0.PSet (IntX, Y), RGB(255, 255, 255)

 Next Y

 'The following section is to find the orientation and displacement
 'Comparing the theoretical zero point and new zero point
 'Calculate the displacement and titled angle

 ZeroO_X = (intUpperBoundX - 1) / 2
 ZeroO_Y = (intUpperBoundY - 1) / 2

 If ((Center_Point_X - ZeroO_X = 0) And (Center_Point_Y - ZeroO_Y = 0)) Then

   Theta = 0
   Displacement = 0

  Else

   Displacement = Sqr((Center_Point_X - ZeroO_X) ^ 2 + (Center_Point_Y - ZeroO_Y) ^ 2)
   TempDouble = (Center_Point_Y - ZeroO_Y) / Displacement
   Theta = Atn(TempDouble / Sqr(-TempDouble * TempDouble + 1))
   Angle = 90 - ((Theta / 3.141592654) * 180)

 End If

 MsgBox ("Titled angle is (clockwise):   " & Angle)
 MsgBox ("Displacement is:   " & Displacement)
```

```
For X = 0 To intUpperBoundX - 1
 For Y = 0 To intUpperBoundY - 1
  Picture0.PSet (X, Y), RGB(255, 255, 255)
  Next Y
Next X

For i = 0 To 6

 Picture0.CurrentX = 20
 Picture0.CurrentY = 20 + 15 * i
 Select Case i
   Case 0:
     Picture0.Print ("Number of center lines are  " & n)
     TempText = "Number of center lines are:   "
     Write #1, TempText, n

   Case 1:
     Picture0.Print ("C.L.Intercept = " & CenterLineIntercept)
     Write #1, "C.L.Intercept = ", CenterLineIntercept

   Case 2:
     Picture0.Print ("C.L.Slope = " & CenterLineSlope)
     Write #1, "C.L.Slope = ", CenterLineSlope

   Case 3:
     Picture0.Print ("Center X = " & Center_Point_X)
     Write #1, "Center X = ", Center_Point_X

   Case 4:
     Picture0.Print ("Center Y = " & Center_Point_Y)
     Write #1, "Center Y = ", Center_Point_Y

   Case 5:
     Picture0.Print ("Titled angle is (clockwise):   " & Angle)
     Write #1, "Titled angle is (clockwise):  ", Angle

   Case 6:
     Picture0.Print ("Displacement is:   " & Displacement)
     Write #1, "Displacement is   ", Displacement

 End Select
Next i
Close #1

MsgBox "Ori & Dis. routine ended !"

End Sub

Private Sub cmdQuit_Click()

cmdQUIT.ToolTipText = "Exit the system"

   Unload Form4
   Exit Sub
  'Form2.Show

End Sub
Private Sub cmdBack_Click()

 cmdBack.ToolTipText = "Back to previous stage"

 Unload Form4
 Image_Capture.Show
```

```
End Sub
Private Sub Dir1_Change()
  File1.Path = Dir1.Path
End Sub
Private Sub Drive1_Change()
  Dir1.Path = Drive1.Drive
End Sub
Function GetRed(colorVal As Long) As Integer
  GetRed = colorVal Mod 256
End Function
Function GetGreen(colorVal As Long) As Integer
  GetGreen = ((colorVal And &HFF00FF00) / 256&)
End Function
Function GetBlue(colorVal As Long) As Integer
  GetBlue = (colorVal And &HFF0000) / (256& * 256&)
End Function
Function GetSlope(Points As Integer, LineN As Integer, Choice As Integer) As Double

  Dim SumXY, SumX, SumY As Double
  Dim SumYsq, SumXsq, FuncDumy As Double
  Dim a, Index, Position_X, Position_Y As Integer

  SumXY = 0
  SumX = 0
  SumY = 0
  SumXsq = 0
  SumYsq = 0
  Position_X = 0
  Position_Y = 0
  Index = 0
  FuncDumy = 0

  'Sometimes the image is trancated; u do not have
  'all the 18 points; we use the B to represent to count
  'all the points
  'Choice 0: Line correlation coefficient
  'Choice 1: Parallel line slope
  'Choice 2: Center line slope
  'Choice 3: Center line intercept

 If (Choice = 0 Or Choice = 1) Then
   For a = 0 To Points - 1
     Position_X = Coord_X(a, LineN)
     Position_Y = Coord_Y(a, LineN)

     If ((Position_X <> 0) And (Position_Y <> 0)) Then
       SumXY = SumXY + (Position_X * Position_Y)
       SumX = SumX + Position_X
       SumY = SumY + Position_Y
       SumYsq = SumYsq + Position_Y ^ 2
       SumXsq = SumXsq + Position_X ^ 2
       Index = Index + 1
     End If
   Next a

 End If

If (Choice = 2 Or Choice = 3) Then
   For a = 0 To Points - 1
    For LineN = 0 To n - 1
     Position_X = Coord_X(a, LineN)
     Position_Y = Coord_Y(a, LineN)

     If ((Position_X <> 0) And (Position_Y <> 0)) Then
       SumXY = SumXY + (Position_X * Position_Y)
       SumX = SumX + Position_X
```

```
        SumY = SumY + Position_Y
        SumYsq = SumYsq + Position_Y ^ 2
        SumXsq = SumXsq + Position_X ^ 2
        Index = Index + 1

      End If
    Next LineN
   Next a
End If

 If ((SumX = 0) Or (SumY = 0) Or (SumX * SumY = 0)) Then
   GetSlope = 0
 Else
   If (Choice = 1 Or Choice = 2) Then
     GetSlope = ((SumXY) - ((SumX * SumY) / Index)) / ((SumYsq) - ((SumY * SumY) / Index))
   End If

   If (Choice = 3) Then
     FuncDumy = ((SumXY) - ((SumX * SumY) / Index)) / ((SumYsq) - ((SumY * SumY) / Index))
     GetSlope = (SumX - (FuncDumy * SumY)) / Index
   End If

   If (Choice = 0) Then
     FuncDumy = Sqr((SumXsq - (SumX ^ 2 / Index)) * (SumYsq - (SumY ^ 2 / Index)))
     GetSlope = ((SumXY) - ((SumX * SumY) / Index)) / FuncDumy

   End If
 End If

End Function
Function dblSquare(SquareMe As Integer) As Double

  dblSquare = SquareMe ^ 2 '* SquareMe

End Function
Function Measure_Distance(c1 As Double, m1 As Double, Point2_X As Integer, Point2_Y As Integer) As Double

Dim Point1_X, Point1_Y As Long
Dim c2 As Long

c2 = Point2_X - ((-1 / m1) * Point2_Y)
Point1_X = (c2 * m1 - c1 * (-1 / m1)) / (m1 - (-1 / m1))
Point1_Y = (c2 - c1) / (m1 - (-1 / m1))
Measure_Distance = Sqr((Point2_X - Point1_X) ^ 2 + (Point2_Y - Point1_Y) ^ 2)

End Function

Private Sub Picture0_Click()

End Sub
```

820F5 FORM

```
Private Sub cmdBack_Click()

cmdBack.ToolTipText = "Back to previous stage"

Unload Form5
Form4.Show

End Sub

Private Sub Picture2_Click()
```

```
End Sub

Private Sub cmdQuit_Click()

cmdQuit.ToolTipText = "Exit from the system"

Unload Form5
Exit Sub

End Sub
Private Sub cmdShowRes_Click()

Dim NewLine As String

cmdShowRes.ToolTipText = "Display the inspection results"

On Error GoTo FileError
Open "c:\windows\desktop\InspResults.txt" For Input As #1
Do Until EOF(1)
  Line Input #1, NewLine
  TEXT1.Text = TEXT1.Text + NewLine + vbCrLf
Loop

Exit Sub

FileError:
 MsgBox "File Error! "

End Sub

Private Sub Form_Load()

End Sub
```

IMAGE_CAPTURE FORM

```
Dim cfg As VPX_Config
Dim hDIB As Long
Dim numAverage As Integer
Dim prevAverageIndex As Integer
Dim filtOn(8) As Boolean
Dim filt(8) As Long
Dim avgNums(6) As Integer
Dim Err As Integer

Private Declare Function GlobalFree Lib "KERNEL32" (ByVal handle&) As Long
Sub SetupMenu()
   Dim Enable As Boolean
   Enable = cfg.outputFormat = VPP_mono Or cfg.outputFormat = VPP_BGR24

   'ImageCaptureFunctionEnable initialized in the password Form

   For i% = 1 To 6
      If (ImageCaptureFunctionsEnable = 0) Then
        Average(i%).Enabled = False
      End If
      If (ImageCaptureFunctionsEnable = 1) Then
        Average(i%).Enabled = True
      End If
   Next i%

   For i% = 1 To 8
      If (ImageCaptureFunctionsEnable = 0) Then
        Filter(i%).Enabled = False
```

```
      End If
      If (ImageCaptureFunctionsEnable = 1) Then
        Filter(i%).Enabled = True 'enable
      End If
    Next i%

    If (ImageCaptureFunctionsEnable = 0) Then
      ImageFormat.Enabled = False
      Copy.Enabled = False
    End If
    If (ImageCaptureFunctionsEnable = 1) Then
      ImageFormat.Enabled = True 'User sh not have aces to ths fun
      Copy.Enabled = True
    End If


  End Sub

  Private Sub Average_Click(Index As Integer)
    Average(prevAverageIndex).Checked = False
    Average(Index).Checked = True
    prevAverageIndex = Index
    numAverage = avgNums(Index)
  End Sub

  Private Sub Copy_Click()

    If hDIB <> 0 Then
      Dim hDIB2 As Long
      Check (VPX_copyDIB(hDIB, hDIB2))
      Check (VPX_saveDIBToClipboard(hDIB2))
    End If
  End Sub

  Private Sub Exit_Click()

   Unload Image_Capture
   Form1Backup.Show


  End Sub

  Private Sub filter_Click(Index As Integer)
    filtOn(Index) = Not filtOn(Index)
    Filter(Index).Checked = filtOn(Index)
  End Sub

  Private Sub Form_Load()

    hDIB = 0
    numAverage = 1
    prevAverageIndex = 1
    For i% = 1 To 8
      filtOn(i%) = False
    Next i%
    filt(1) = VPX_AVERAGE
    filt(2) = VPX_SMOOTH
    filt(3) = VPX_DETAIL
    filt(4) = VPX_SHARPEN
    filt(5) = VPX_AI
    filt(6) = VPX_AISHARPEN
    filt(7) = VPX_VERTLINES
    filt(8) = VPX_HORIZLINES
    avgNums(1) = 1
    avgNums(2) = 2
    avgNums(3) = 3
```

```
    avgNums(4) = 4
    avgNums(5) = 8
    avgNums(6) = 16
    Check (VPP_init())
    Err = VPX_readIniFile(".\test.ini", "DEFAULT", cfg)
    Check (VPX_prepare(cfg, VPP_true))
    If cfg.outputFormat = VPP_mono Or cfg.outputFormat = VPP_mono4 Then
        Err = VPX_defaultPalette(Image_Capture.hDC, VPP_true)
    Else
        Err = VPX_defaultPalette(Image_Capture.hDC, VPP_false)
    End If
    SetupMenu
    Timer1.Enabled = True
End Sub

Private Sub Form_Unload(Cancel As Integer)
    If hDIB <> 0 Then
        GlobalFree (hDIB)
    End If
    Timer1.Enabled = False
    Check (VPX_saveIniFile(".\test.ini", "DEFAULT", cfg))
    Err% = VPX_releasePalette()
    Check (VPP_closedown(VPP_true))
End Sub

Private Sub ImageFormat_Click()

    Timer1.Enabled = False
    Err = VPX_formatDialogBox(0, 0, cfg)
    Err = VPX_prepare(cfg, VPP_true)
    SetupMenu
    Timer1.Enabled = True
End Sub

Private Sub Save_Click()
    If hDIB <> 0 Then
        Check (VPX_saveDIBToFile(hDIB, ".\test.bmp"))
    End If

End Sub

Private Sub Timer1_Timer()
    Dim formatOk As Boolean
    Timer1.Enabled = False
    If hDIB <> 0 Then
        handle& = GlobalFree(hDIB)
    End If
    formatOk = cfg.outputFormat = VPP_mono Or cfg.outputFormat = VPP_BGR24
    If numAverage > 1 And formatOk Then
        Err = VPX_snapAverageDIB(cfg, hDIB, numAverage, numAverage)
    Else
        Err = VPX_snap(cfg)
        Err = VPX_readoutDIB(cfg, hDIB)
    End If
    For i% = 1 To 8
        If filtOn(i%) And formatOk Then
            Err = VPX_filterDIBPredef(hDIB, VPX_getFilter(filt(i%)))
        End If
    Next i%
    Err = VPX_drawDIB(Image_Capture.hDC, hDIB, 0, 0, 0, 0)

    'Dim intLoopIndex As Integer
    'For intLoopIndex = 0 To 17
    '  Line (1000, 1000 + 400 * intLoopIndex)-(3500, 1000 + 400 _
    '  * intLoopIndex), RGB(255, 255, 0)
    'Next intLoopIndex
```

```
    The following is the boundary of the image

    Line (220, 7550)-(11400, 7550), RGB(255, 255, 0) 'Bot.Hoz.
    Line (220, 280)-(11400, 280), RGB(255, 255, 0)   'Top Hoz.
    Line (220, 7550)-(220, 280), RGB(255, 255, 0)    'Left Ver.
    Line (11400, 7550)-(11400, 280), RGB(255, 255, 0) 'Rgt Ver.

    The following is the cross-hair of the area

    Line (5675, 3913)-(5975, 3913), RGB(255, 255, 0) 'Horizontal line
    Line (5825, 4013)-(5825, 3813), RGB(255, 255, 0) 'Verticle line

    Timer1.Enabled = True
End Sub
```

820PASSWORD FORM

```
Private Sub cmdLogin_Click()

If txtpasswd.Text = "password" Then
Unload password

'Initialize the settigns

DummyY = 0
ImageCaptureFunctionsEnable = 0

Form2.Show
MsgBox "Please set up the HMD first !"

Else
 MsgBox "Wrong Passord ! Please enter again !"


End If

End Sub

Private Sub cmdRestart_Click()
txtpasswd.Text = ""
End Sub

Private Sub cmdQuit_Click()
Unload password
exitwnd.Show
End Sub

Private Sub Form_Load()
txtpasswd.Text = ""
End Sub

Private Sub Image2_Click()

End Sub
```

PASSWORDFORSETTINGS FORM

```
Private Sub Command1_Click()
If txtpasswd.Text = "passwordforsettings" Then
specs.Show
specs!TEXT1.Text = ""
specs!Text2.Text = ""
specs!Text3.Text = ""
```

```
specs!Text4.Text = ""
Unload passwordforsettings
Else
 MsgBox "Wrong Passard ! Please enter again !"

End If
End Sub

Private Sub Command2_Click()
txtpasswd.Text = ""
End Sub

Private Sub Command3_Click()

Unload passwordforsettings
exitwnd.Show

End Sub

Private Sub Form_Load()
txtpasswd.Text = ""
End Sub

Private Sub Image2_Click()

End Sub
```

SETTINGS FORM

```
Private Sub end_Click()

Unload Me
Form2.Show

End Sub

Private Sub Image2_Click()

End Sub

Private Sub Reset_Click()

Unload Me
specs.Show

End Sub
```

SPECTEST FORM

```
Private Sub Command1_Click()

settings.Show
settings!Text1.Text = specs!Text1.Text 'Height
settings!Text2.Text = specs!Text2.Text 'W2
settings!Text3.Text = specs!Text3.Text 'Width
settings!Text4.Text = specs!Text4.Text 'W1

'The following are the Public variables
'Declared in the Image_Capture_Module
'Val() convert the string into integer
```

```
PatternWidth = Val(Text3.Text)
PatternHeight = Val(Text1.Text)
PatternW1 = Val(Text4.Text)
PatternW2 = Val(Text2.Text)

'The following are the testing routines
'Height = Val(Text1.Text)
'MsgBox ("Height is " & Str(PatternHeight))
'MsgBox ("Width is " & Str(PatternWidth))
'MsgBox ("W1 is " & Str(PatternW1))
'MsgBox ("W2 is " & Str(PatternW2))

Unload specs
End Sub

Private Sub Disable_Click()

ImageCaptureFunctionsEnable = 0

End Sub

Private Sub Enable_Click()

ImageCaptureFunctionsEnable = 1

End Sub

Private Sub Image2_Click()

End Sub
```

IMAGE_CAPTURE MODULE

```
Public ImageCaptureFunctionsEnable As Integer
Public PatternWidth, PatternHeight, PatternW1, PatternW2 As Integer

Public DummyY As Integer
Public FlagLabel As Integer

'Bool
Global Const VPP_false = 0
Global Const VPP_true = 1

'Error numbers
Global Const VPP_success = 0                'No error
Global Const VPP_toolkitInUse = 1           'VideoPort toolkit is already in use
Global Const VPP_noHardwareDetected = 2      'No VideoPort hardware detected
Global Const VPP_noDriverDetected = 3        'No VideoPort PCMCIA driver detected
Global Const VPP_oldVideoPortDetected = 4    'The installed VideoPort is old-style
Global Const VPP_notInitialized = 5          'init        has not been called
Global Const VPP_notConfigured = 6           'videoConfig has not been called
Global Const VPP_snapNotPrepared = 7         'prepareSnap has not been called
Global Const VPP_snapNotStarted = 8          'startSnap   has not been called
Global Const VPP_snapNotFinished = 9         'finishSnap  has not been called
Global Const VPP_readoutNotStarted = 10      'startReadout has not been called
Global Const VPP_noSignalDetected = 11       'No video signal detected
Global Const VPP_noColorSnapped = 12         'Snapped image does not contain colour
Global Const VPP_readoutOutsideSnappedImage = 13 'Attempt to read outsize snapped image
Global Const VPP_parameterOutOfRange = 14    'Parameter to function is out of range
Global Const VPP_imageWidthOutOfRange = 15   'Image width is out of range
Global Const VPP_imageHeightOutOfRange = 16  'Image height is out of range
Global Const VPP_badPointer = 17             'Bad pointer (possibly NULL)
Global Const VPP_lostContact = 18            'Contact with VideoPort is lost
Global Const VPP_outOfMemory = 19            'Could not claim the memory needed
```

```
Global Const VPP_fileIOError = 20          'File input/output error

'Global constants
Global Const VPP_DEFAULT_CHANNEL = 0
Global Const VPP_MIN_BRIGHTNESS = -128
Global Const VPP_MIN_CONTRAST = -128
Global Const VPP_MIN_SATURATION = -128
Global Const VPP_MIN_HUE = -128
Global Const VPP_DEFAULT_BRIGHTNESS = 0
Global Const VPP_DEFAULT_CONTRAST = 0
Global Const VPP_DEFAULT_SATURATION = 0
Global Const VPP_DEFAULT_HUE = 0
Global Const VPP_DEFAULT_GAMMA = 1
Global Const VPP_MAX_BRIGHTNESS = 127
Global Const VPP_MAX_CONTRAST = 127
Global Const VPP_MAX_SATURATION = 127
Global Const VPP_MAX_HUE = 127
Global Const VPP_DEFAULT_FLASH_DELAY = 8

'Video standards
Global Const VPP_NTSC = 0
Global Const VPP_PAL = 1
Global Const VPP_noSignal = 2

'Signal types
Global Const VPP_composite = 0
Global Const VPP_Svideo = 1
Global Const VPP_monochrome = 2

Type VPP_SnapData
  xOffset As Integer   'X offset of active video area
  xActive As Integer   'Width of active video area
  xPixels As Integer   'Requested width in pixels of active video area
  yOffset As Integer   'Y offset of active video area
  yActive As Integer   'Height of active video area
  yPixels As Integer   'Requested height in pixels of active video area
  interlaced As Long   'Flag to turn on interlaced snap
  monochrome As Long   'Flag to turn on monochrome snap
End Type

Type VPP_LimitData
  xActiveMax As Integer   'PAL: 922, NTSC: 754
  xPixelsMax As Integer   'PAL: 922, NTSC: 754, VideoPort Junior: 510
  xActiveRatio As Integer 'Currently: 14
  xPixelsRatio As Integer 'Currently: 1
  yActiveMax As Integer   'PAL: 610, NTSC: 510
  yPixelsMax As Integer   'PAL: 610, NTSC: 510, VideoPort Junior: 510
  yActiveRatio As Integer 'Currently: 14
  yPixelsRatio As Integer 'Currently: 1
End Type

'Readout modes        'Format:                    Size factor:
Global Const VPP_mono = 1     'Byte I...                  1
Global Const VPP_mono4 = 2    'Nibble IIII(2)... dithered    1/2
Global Const VPP_RGB8 = 3     'Byte RRRGGGBB(2)... dithered    1
Global Const VPP_RGB15 = 4    'Word 0RRRRRGG GGGBBBBB(2)...    2
Global Const VPP_RGB16 = 5    'Word RRRRRGGG GGGBBBBB(2)...    2
Global Const VPP_BGR24 = 7    'Byte B,G,R...              3
Global Const VPP_BGR032 = 8   'Byte B,G,R,0...            4

Declare Function VPP_init Lib "VPX32.DLL" () As Long
Declare Function VPP_closedown Lib "VPX32.DLL" (ByVal powerOff&) As Long
Declare Function VPP_getCurrentCardHandle Lib "VPX32.DLL" (cardhandle&) As Long
Declare Function VPP_setActiveCard Lib "VPX32.DLL" (ByVal cardhandle&) As Long
Declare Function VPP_videoConfig Lib "VPX32.DLL" (ByVal channel%, ByVal signalType&, videoStandard&)
As Long
```

```
Declare Function VPP_testSignal Lib "VPX32.DLL" (videoStandard&) As Long
Declare Function VPP_setBrightness Lib "VPX32.DLL" (ByVal brightness%) As Long
Declare Function VPP_setContrast Lib "VPX32.DLL" (ByVal contrast%) As Long
Declare Function VPP_setSaturation Lib "VPX32.DLL" (ByVal saturation%) As Long
Declare Function VPP_setHue Lib "VPX32.DLL" (ByVal hue%) As Long
Declare Function VPP_setGamma Lib "VPX32.DLL" (ByVal gamma!) As Long
Declare Function VPP_enableFlash Lib "VPX32.DLL" (ByVal flashSelect&, ByVal mustBeNULL&, ByVal
flashDelay%) As Long
Declare Function VPP_disableFlash Lib "VPX32.DLL" () As Long
Declare Function VPP_getLimits Lib "VPX32.DLL" (ByVal videoStandard&, limitData As VPP_LimitData) As
Long
Declare Function VPP_prepareSnap Lib "VPX32.DLL" (snapData As VPP_SnapData) As Long
Declare Function VPP_startSnap Lib "VPX32.DLL" () As Long
Declare Function VPP_finishSnap Lib "VPX32.DLL" () As Long
Declare Function VPP_flashSnap Lib "VPX32.DLL" () As Long
Declare Function VPP_extTrigSnap Lib "VPX32.DLL" (ByVal msecTimeout%) As Long
Declare Function VPP_autoCrop Lib "VPX32.DLL" (snapData As VPP_SnapData) As Long

Type VPX_Config
 videoStandard As Long
 signalType As Long
 inputChannel As Long
 brightness As Long
 contrast As Long
 saturation As Long
 gamma As Single
 hue As Long
 snapDataNTSC As VPP_SnapData
 snapDataPAL As VPP_SnapData
 outputFormat As Long
 extTrigSnap As Long
 flashSnap As Long
 extTrigEnable As Long
 flashEnable As Long
 mono4Enable As Long
 monoEnable As Long
 RGB8Enable As Long
 BGR24Enable As Long
 RGB15Enable As Long
 RGB16Enable As Long
 BGR032Enable As Long
End Type

Type VPX_Filter
 Width As Byte
 Height As Byte
 divideBy As Long
 doAbs As Long
 'The following scheme allows up to 4x4 or 3x5 filters
 coeff0 As Byte      'Treat with care, should be "signed char"
 coeff1 As Byte      'Treat with care, should be "signed char"
 coeff2 As Byte      'Treat with care, should be "signed char"
 coeff3 As Byte      'Treat with care, should be "signed char"
 coeff4 As Byte      'Treat with care, should be "signed char"
 coeff5 As Byte      'Treat with care, should be "signed char"
 coeff6 As Byte      'Treat with care, should be "signed char"
 coeff7 As Byte      'Treat with care, should be "signed char"
 coeff8 As Byte      'Treat with care, should be "signed char"
 coeff9 As Byte      'Treat with care, should be "signed char"
 coeff10 As Byte      'Treat with care, should be "signed char"
 coeff11 As Byte      'Treat with care, should be "signed char"
 coeff12 As Byte      'Treat with care, should be "signed char"
 coeff13 As Byte      'Treat with care, should be "signed char"
 coeff14 As Byte      'Treat with care, should be "signed char"
 coeff15 As Byte      'Treat with care, should be "signed char"
End Type
```

```
'Predefined filters
Global Const VPX_AVERAGE = 1
Global Const VPX_SMOOTH = 2
Global Const VPX_DETAIL = 3
Global Const VPX_SHARPEN = 4
Global Const VPX_AI = 5
Global Const VPX_AISHARPEN = 6
Global Const VPX_VERTLINES = 7
Global Const VPX_HORIZLINES = 8

Declare Function VPX_defaultConfig Lib "VPX32.DLL" (config As VPX_Config) As Long
Declare Function VPX_prepare Lib "VPX32.DLL" (config As VPX_Config, ByVal forceConfig&) As Long
Declare Function VPX_snap Lib "VPX32.DLL" (config As VPX_Config) As Long
Declare Function VPX_draw Lib "VPX32.DLL" (ByVal hDC&, config As VPX_Config, ByVal X%, ByVal Y%,
ByVal x1%, ByVal y2%) As Long
Declare Function VPX_drawDIB Lib "VPX32.DLL" (ByVal hDC&, ByVal hDIB&, ByVal X%, ByVal Y%,
ByVal x1%, ByVal y2%) As Long
Declare Function VPX_readoutDIB Lib "VPX32.DLL" (config As VPX_Config, hDIB&) As Long
Declare Function VPX_readoutDIBToClipboard Lib "VPX32.DLL" (config As VPX_Config) As Long
Declare Function VPX_saveDIBToClipboard Lib "VPX32.DLL" (ByVal hDIB&) As Long
Declare Function VPX_readoutDIBToFile Lib "VPX32.DLL" (config As VPX_Config, ByVal filename$) As
Long
Declare Function VPX_saveDIBToFile Lib "VPX32.DLL" (ByVal hDIB&, ByVal filename$) As Long
Declare Function VPX_snapAverageDIB Lib "VPX32.DLL" (config As VPX_Config, hDIB&, ByVal
numAverage%, ByVal divideBy%) As Long
Declare Function VPX_filterDIB Lib "VPX32.DLL" (ByVal hDIB&, Filter As VPX_Filter) As Long
Declare Function VPX_filterDIBPredef Lib "VPX32.DLL" Alias "VPX_filterDIB" (ByVal hDIB&, ByVal
Filter&) As Long
Declare Function VPX_getFilter Lib "VPX32.DLL" (ByVal filterNo&) As Long
Declare Function VPX_copyDIB Lib "VPX32.DLL" (ByVal hDIB&, phDIB&) As Long
Declare Function VPX_readIniFile Lib "VPX32.DLL" (ByVal File$, ByVal section$, config As VPX_Config) As
Long
Declare Function VPX_saveIniFile Lib "VPX32.DLL" (ByVal File$, ByVal section$, config As VPX_Config) As
Long
Declare Function VPX_defaultPalette Lib "VPX32.DLL" (ByVal hDC&, ByVal monochrome&) As Long
Declare Function VPX_releasePalette Lib "VPX32.DLL" () As Long
Declare Function VPX_formatDialogBox Lib "VPX32.DLL" (ByVal ignored&, ByVal parent&, config As
VPX_Config) As Long

Sub Check(ret As Integer)
  If ret <> VPP_success Then
    MsgBox "Error returned from VPPTOOLS: " + Chr$(10) + VPP_errorString(ret), MB_OK +
MB_ICONSTOP, "Test application"
    ret = VPP_closedown(VPP_true)
    End
  End If
End Sub

Function VPP_errorString(errno As Integer) As String
  Select Case errno
    Case VPP_success
      VPP_errorString = "No error"
    Case VPP_toolkitInUse
      VPP_errorString = "VideoPort toolkit is already in use"
    Case VPP_noHardwareDetected
      VPP_errorString = "No VideoPort hardware detected"
    Case VPP_noDriverDetected
      VPP_errorString = "No VideoPort PCMCIA driver detected"
    Case VPP_oldVideoPortDetected
      VPP_errorString = "The installed VideoPort is old-style"
    Case VPP_notInitialized
      VPP_errorString = "init has not been called"
    Case VPP_notConfigured
      VPP_errorString = "videoConfig has not been called"
    Case VPP_snapNotPrepared
```

```
      VPP_errorString = "prepareSnap has not been called"
    Case VPP_snapNotStarted
      VPP_errorString = "startSnap has not been called"
    Case VPP_snapNotFinished
      VPP_errorString = "finishSnap has not been called"
    Case VPP_readoutNotStarted
      VPP_errorString = "startReadout has not been called"
    Case VPP_noSignalDetected
      VPP_errorString = "No video signal detected"
    Case VPP_noColorSnapped
      VPP_errorString = "Snapped image does not contain colour"
    Case VPP_readoutOutsideSnappedImage
      VPP_errorString = "Attempt to read outsize snapped image"
    Case VPP_parameterOutOfRange
      VPP_errorString = "Parameter to function is out of range"
    Case VPP_imageWidthOutOfRange
      VPP_errorString = "Image width is out of range"
    Case VPP_imageHeightOutOfRange
      VPP_errorString = "Image height is out of range"
    Case VPP_badPointer
      VPP_errorString = "Bad pointer (possibly NULL)"
    Case VPP_lostContact
      VPP_errorString = "Contact with VideoPort is lost"
    Case VPP_outOfMemory
      VPP_errorString = "Out of memory"
    Case VPP_fileIOError
      VPP_errorString = "File I/O error"
    Case Else
      VPP_errorString = "Unknown error"
  End Select
End Function
Public Function Mean(X As Integer, Y As Integer)


End Function
Public Function Center_Point(X As Integer, Y As Integer)

End Function
```

NIDAQ32 MODULE

```
'***************************************************************************
Declare Function AI_Change_Parameter% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c&, ByVal d&)
Declare Function AI_Check% Lib "nidaq32.dll" (ByVal a%, b%, c%)
Declare Function AI_Clear% Lib "nidaq32.dll" (ByVal a%)
Declare Function AI_Configure% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%,
ByVal f%)
Declare Function AI_Mux_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%)
Declare Function AI_Read% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d%)
Declare Function AI_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function AI_VRead% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d#)
Declare Function AI_VScale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, ByVal e#, ByVal
f%, g#)
Declare Function Align_DMA_Buffer% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c As Any, ByVal d&, ByVal
e&, f&)
Declare Function AO_Calibrate% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function AO_Configure% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e#,
ByVal f%)
Declare Function AO_Change_Parameter% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c&, ByVal d&)
Declare Function AO_Update% Lib "nidaq32.dll" (ByVal a%)
```

```
Declare Function AO_VWrite% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c#)
Declare Function AO_Write% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function Calibrate_E_Series% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d#)
Declare Function Calibrate_59xx% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c#)
Declare Function Calibrate_DSA% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c#)
Declare Function Config_Alarm_Deadband% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c$, ByVal d#,
ByVal e#, ByVal f%, ByVal g%, ByVal h%, ByVal i&)
Declare Function Config_ATrig_Event_Message% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c$, ByVal d#,
ByVal e#, ByVal f%, ByVal g&, ByVal h&, ByVal i&, ByVal j%, ByVal k%, ByVal l&)
Declare Function Config_DAQ_Event_Message% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c$, ByVal
d%, ByVal e&, ByVal f&, ByVal g&, ByVal h&, ByVal i&, ByVal j%, ByVal k%, ByVal l&)
Declare Function Configure_HW_Analog_Trigger% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal
d&, ByVal e&, ByVal f&)
Declare Function CTR_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%,
ByVal f%)
Declare Function CTR_EvCount% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
Declare Function CTR_EvRead% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%)
Declare Function CTR_FOUT_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal
e%)
Declare Function CTR_Period% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function CTR_Pulse% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)
Declare Function CTR_Rate% Lib "nidaq32.dll" (ByVal a#, ByVal b#, c%, d%, e%)
Declare Function CTR_Reset% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function CTR_Restart% Lib "nidaq32.dll" (ByVal a%, ByVal b%)
Declare Function CTR_Simul_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, ByVal d%)
Declare Function CTR_Square% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)
Declare Function CTR_State% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)
Declare Function CTR_Stop% Lib "nidaq32.dll" (ByVal a%, ByVal b%)
Declare Function DAQ_Check% Lib "nidaq32.dll" (ByVal a%, b%, c&)
Declare Function DAQ_Clear% Lib "nidaq32.dll" (ByVal a%)
Declare Function DAQ_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function DAQ_DB_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%)
Declare Function DAQ_DB_HalfReady% Lib "nidaq32.dll" (ByVal a%, b%, c%)
Declare Function DAQ_DB_Transfer% Lib "nidaq32.dll" (ByVal a%, b As Any, c&, d%)
Declare Function DAQ_Monitor% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d&, e As Any, f&,
g%)
Declare Function DAQ_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d As Any, ByVal e&, ByVal
f#)
Declare Function DAQ_Rate% Lib "nidaq32.dll" (ByVal a#, ByVal b%, c%, d%)
Declare Function DAQ_Start% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d As Any, ByVal e&, ByVal
f%, ByVal g%)
Declare Function DAQ_StopTrigger_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c&)
Declare Function DAQ_to_Disk% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d$, ByVal e&,
ByVal f#, ByVal g%)
Declare Function DAQ_VScale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, ByVal e#,
ByVal f&, g%, h#)
Declare Function DIG_Block_Check% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c&)
Declare Function DIG_Block_Clear% Lib "nidaq32.dll" (ByVal a%, ByVal b%)
Declare Function DIG_Block_In% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c As Any, ByVal d&)
Declare Function DIG_Block_Out% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c As Any, ByVal d&)
Declare Function DIG_Block_PG_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%,
ByVal e%, ByVal f%, ByVal g%)
Declare Function DIG_DB_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)
Declare Function DIG_DB_HalfReady% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)
Declare Function DIG_DB_Transfer% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c As Any, ByVal d&)
Declare Function DIG_Grp_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)
Declare Function DIG_Grp_Mode% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%,
ByVal f%, ByVal g%)
Declare Function DIG_Grp_Status% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)
Declare Function DIG_In_Grp% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)
Declare Function DIG_In_Line% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d%)
Declare Function DIG_In_Port% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)
Declare Function DIG_Line_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
Declare Function DIG_Out_Grp% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function DIG_Out_Line% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
Declare Function DIG_Out_Port% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
```

```
Declare Function DIG_Prt_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
Declare Function DIG_Prt_Status% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)
Declare Function DIG_SCAN_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d%, ByVal e%)
Declare Function Get_DAQ_Device_Info% Lib "nidaq32.dll" (ByVal a%, ByVal b&, c&)
Declare Function Get_DAQ_Event% Lib "nidaq32.dll" (ByVal a&, b%, c%, d%, e&)
Declare Function Get_NI_DAQ_Version% Lib "nidaq32.dll" (a&)
Declare Function GPCTR_Config_Buffer% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d&, e As Any)
Declare Function GPCTR_Read_Buffer% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d&, ByVal e&, ByVal f#, g&, h&)
Declare Function Line_Change_Attribute% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d&)
Declare Function GPCTR_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&)
Declare Function GPCTR_Set_Application% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&)
Declare Function GPCTR_Watch% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, d&)
Declare Function ICTR_Read% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)
Declare Function ICTR_Reset% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function ICTR_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)
Declare Function Init_DA_Brds% Lib "nidaq32.dll" (ByVal a%, b%)
Declare Function Lab_ISCAN_Check% Lib "nidaq32.dll" (ByVal a%, b%, c&, d%)
Declare Function Lab_ISCAN_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d As Any, ByVal e&, ByVal f#, ByVal g#, h%)
Declare Function Lab_ISCAN_Start% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d As Any, ByVal e&, ByVal f%, ByVal g%, ByVal h%)
Declare Function Lab_ISCAN_to_Disk% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d$, ByVal e&, ByVal f#, ByVal g#, ByVal h%)
Declare Function LPM16_Calibrate% Lib "nidaq32.dll" (ByVal a%)
Declare Function MIO_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function Peek_DAQ_Event% Lib "nidaq32.dll" (ByVal a&, b%, c%, d%, e&)
Declare Function REG_Level_Read% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c&)
Declare Function REG_Level_Write% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c&, ByVal d&, e&)
Declare Function RTSI_Clear% Lib "nidaq32.dll" (ByVal a%)
Declare Function RTSI_Clock% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function RTSI_Conn% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
Declare Function RTSI_DisConn% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function SC_2040_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function SCAN_Demux% Lib "nidaq32.dll" (a%, ByVal b&, ByVal c%, ByVal d%)
Declare Function SCAN_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%, e As Any, ByVal f&, ByVal g#, ByVal h#)
Declare Function SCAN_Sequence_Demux% Lib "nidaq32.dll" (ByVal a%, b%, ByVal c&, d%, ByVal e%, f%, g&)
Declare Function SCAN_Sequence_Retrieve% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)
Declare Function SCAN_Sequence_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%, e%, f%, g%)
Declare Function SCAN_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%)
Declare Function SCAN_Start% Lib "nidaq32.dll" (ByVal a%, b As Any, ByVal c&, ByVal d%, ByVal e%, ByVal f%, ByVal g%)
Declare Function SCAN_to_Disk% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%, ByVal e$, ByVal f&, ByVal g#, ByVal h#, ByVal i%)
Declare Function Calibrate_1200% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f%, ByVal g%, ByVal h%, ByVal i#, ByVal j#)
Declare Function SCXI_AO_Write% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f#, ByVal g%, h%)
Declare Function SCXI_Cal_Constants% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f%, ByVal g#, ByVal h%, ByVal i%, ByVal j%, ByVal k#, ByVal l#, ByVal m#, ByVal n#, ByVal o#, p#, q#)
Declare Function SCXI_Calibrate% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f#, ByVal g#, ByVal h%, ByVal i%)
Declare Function SCXI_Calibrate_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function SCXI_Change_Chan% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function SCXI_Set_Excitation% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e!, f!)
Declare Function SCXI_Configure_Connection% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
Declare Function SCXI_Configure_Filter% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e#, ByVal f%, ByVal g%, h#)
Declare Function SCXI_Get_Chassis_Info% Lib "nidaq32.dll" (ByVal a%, b%, c%, d%, e%, f%)
Declare Function SCXI_Get_Module_Info% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c&, d%, e%)
```

```
Declare Function SCXI_Get_State% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, e&)
Declare Function SCXI_Get_Status% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d&)
Declare Function SCXI_Load_Config% Lib "nidaq32.dll" (ByVal a%)
Declare Function SCXI_MuxCtr_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
Declare Function SCXI_Reset% Lib "nidaq32.dll" (ByVal a%, ByVal b%)
Declare Function SCXI_Scale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, ByVal e#, ByVal f%, ByVal g%, ByVal h%, ByVal i&, j%, k#)
Declare Function SCXI_SCAN_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%, e%, ByVal f%, ByVal g%)
Declare Function SCXI_Set_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f%, g&, h%, i%)
Declare Function SCXI_Set_Gain% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#)
Declare Function SCXI_Set_Input_Mode% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function SCXI_Set_State% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e&)
Declare Function SCXI_Single_Chan_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
Declare Function SCXI_Track_Hold_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
Declare Function SCXI_Track_Hold_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f%, ByVal g%)
Declare Function Select_Signal% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d&)
Declare Function Set_DAQ_Device_Info% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&)
Declare Function Timeout_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b&)
Declare Function WFM_Chan_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function WFM_Check% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d&, e&)
Declare Function WFM_ClockRate% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e&, ByVal f%)
Declare Function WFM_DB_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, ByVal d%, ByVal e%, ByVal f%)
Declare Function WFM_DB_HalfReady% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%)
Declare Function WFM_DB_Transfer% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d As Any, ByVal e&)
Declare Function WFM_from_Disk% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, ByVal d$, ByVal e&, ByVal f&, ByVal g&, ByVal h#)
Declare Function WFM_Group_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function WFM_Group_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, ByVal d%)
Declare Function WFM_Load% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d As Any, ByVal e&, ByVal f&, ByVal g%)
Declare Function WFM_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d As Any, ByVal e&, ByVal f&, ByVal g#)
Declare Function WFM_Rate% Lib "nidaq32.dll" (ByVal a#, ByVal b%, c%, d&)
Declare Function WFM_Scale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c&, ByVal d#, e#, f%)
Declare Function AI_Read_Scan% Lib "nidaq32.dll" (ByVal a%, b%)
Declare Function AI_VRead_Scan% Lib "nidaq32.dll" (ByVal a%, b#)
Declare Function SCXI_ModuleID_Read% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c&)
Declare Function USE_E_Series% Lib "nidaq32.dll" ()
Declare Function USE_E_Series_AI% Lib "nidaq32.dll" ()
Declare Function USE_E_Series_AO% Lib "nidaq32.dll" ()
Declare Function USE_E_Series_DIO% Lib "nidaq32.dll" ()
Declare Function USE_E_Series_GPCTR% Lib "nidaq32.dll" ()
Declare Function USE_E_Series_GPCTR_Simple% Lib "nidaq32.dll" ()
Declare Function USE_E_Series_Misc% Lib "nidaq32.dll" ()
Declare Function USE_E_Series_WFM% Lib "nidaq32.dll" ()
Declare Function AO_VScale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c#, d%)
Declare Function GPCTR_Change_Parameter% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d&)
Declare Function USE_E_Series_DAQ% Lib "nidaq32.dll" ()
Declare Function USE_MIO% Lib "nidaq32.dll" ()
Declare Function USE_LPM% Lib "nidaq32.dll" ()
Declare Function USE_LAB% Lib "nidaq32.dll" ()
Declare Function USE_DIO_96% Lib "nidaq32.dll" ()
Declare Function USE_DIO_32F% Lib "nidaq32.dll" ()
Declare Function USE_DIO_24% Lib "nidaq32.dll" ()
Declare Function USE_AO_610% Lib "nidaq32.dll" ()
Declare Function USE_AO_2DC% Lib "nidaq32.dll" ()
Declare Function DIG_Trigger_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f%, ByVal g&, ByVal h&, ByVal i&)
Declare Function SCXI_Set_Threshold% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, ByVal e#)
```

```
Declare Function WFM_Set_Clock% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c&, ByVal d#, ByVal e&, f#)
Declare Function DAQ_Set_Clock% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c#, ByVal d&, e#)
Declare Function Tio_Select_Signal% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d&)
Declare Function Tio_Combine_Signals% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&)
Declare Function DIG_In_Prt% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c&)
Declare Function DIG_Out_Prt% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c&)
Declare Function AI_Get_Overloaded_Channels% Lib "nidaq32.dll" (ByVal a%, b%, c%)
Declare Function Calibrate_TIO% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d#)
Declare Function DIG_Change_Message_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c$, ByVal d$, ByVal e%, ByVal f%, ByVal g&)
Declare Function DIG_Change_Message_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b%)
Declare Function DIG_Filter_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c$, ByVal d#)
```

NIDAQCNS.INC MODULE

```
'**********************************************************************'
'*                                                                    *'
'* This file contains definitions for constants required for some    *'
'* of the NI-DAQ functions.                                           *'
'*                                                                    *'
'* You should use symbols defined here in your programs;  do not     *'
'* use the numerical values.                                          *'
'*                                                                    *'
'* See your NI-DAQ Function Reference Manual for details concerning  *'
'* use of constants defined here.                                     *'
'*                                                                    *'
'**********************************************************************'
Global Const ND_ABOVE_HIGH_LEVEL& = 11020
Global Const ND_AC& = 11025
Global Const ND_ACK_REQ_EXCHANGE_GR1& = 11030
Global Const ND_ACK_REQ_EXCHANGE_GR2& = 11035
Global Const ND_ACTIVE& = 11037
Global Const ND_ADC_RESOLUTION& = 11040
Global Const ND_AI_CALDAC_COUNT& = 11050
Global Const ND_AI_CHANNEL_COUNT& = 11060
Global Const ND_AI_COUPLING& = 11055
Global Const ND_AI_FIFO_INTERRUPTS& = 11600
Global Const ND_ANALOG_FILTER& = 11065
Global Const ND_AO48XDC_SET_POWERUP_STATE& = 42100
Global Const ND_AO_CALDAC_COUNT& = 11070
Global Const ND_AO_CHANNEL_COUNT& = 11080
Global Const ND_AO_EXT_REF_CAPABLE& = 11090
Global Const ND_AO_UNIPOLAR_CAPABLE& = 11095
Global Const ND_ARM& = 11100
Global Const ND_ARMED& = 11200
Global Const ND_ATC_OUT& = 11250
Global Const ND_ATTENUATION& = 11260
Global Const ND_AUTOINCREMENT_COUNT& = 11300
Global Const ND_AUTOMATIC& = 11400
Global Const ND_AVAILABLE_POINTS& = 11500

Global Const ND_BASE_ADDRESS& = 12100
Global Const ND_BELOW_LOW_LEVEL& = 12130
Global Const ND_BOARD_CLOCK& = 12170
Global Const ND_BUFFERED_EVENT_CNT& = 12200
Global Const ND_BUFFERED_PERIOD_MSR& = 12300
Global Const ND_BUFFERED_PULSE_WIDTH_MSR& = 12400
Global Const ND_BUFFERED_SEMI_PERIOD_MSR& = 12500
Global Const ND_BURST& = 12600
Global Const ND_BURST_INTERVAL& = 12700

Global Const ND_CAL_CONST_AUTO_LOAD& = 13050
Global Const ND_CALIBRATION_ENABLE& = 13055
```

```
Global Const ND_CALIBRATION_FRAME_SIZE& = 13060
Global Const ND_CALIBRATION_FRAME_PTR& = 13065
Global Const ND_CJ_TEMP% = &H8000
Global Const ND_CALGND% = &H8001
Global Const ND_CLEAN_UP& = 13100
Global Const ND_CLOCK_REVERSE_MODE_GR1& = 13120
Global Const ND_CLOCK_REVERSE_MODE_GR2& = 13130
Global Const ND_CONFIG_MEMORY_SIZE& = 13150
Global Const ND_CONTINUOUS& = 13160
Global Const ND_COUNT& = 13200

Global Const ND_COUNTER_0& = 13300
Global Const ND_COUNTER_1& = 13400
Global Const ND_COUNTER_2& = 13310
Global Const ND_COUNTER_3& = 13320
Global Const ND_COUNTER_4& = 13330
Global Const ND_COUNTER_5& = 13340
Global Const ND_COUNTER_6& = 13350
Global Const ND_COUNTER_7& = 13360

Global Const ND_COUNTER_1_SOURCE& = 13430
Global Const ND_COUNT_AVAILABLE& = 13450
Global Const ND_COUNT_DOWN& = 13465
Global Const ND_COUNT_UP& = 13485
Global Const ND_COUNT_1& = 13500
Global Const ND_COUNT_2& = 13600
Global Const ND_COUNT_3& = 13700
Global Const ND_COUNT_4& = 13800
Global Const ND_CURRENT_OUTPUT& = 40200

Global Const ND_DAC_RESOLUTION& = 13950
Global Const ND_DATA_TRANSFER_CONDITION& = 13960
Global Const ND_DATA_XFER_MODE_AI& = 14000
Global Const ND_DATA_XFER_MODE_AO_GR1& = 14100
Global Const ND_DATA_XFER_MODE_AO_GR2& = 14200
Global Const ND_DATA_XFER_MODE_DIO_GR1& = 14300
Global Const ND_DATA_XFER_MODE_DIO_GR2& = 14400
Global Const ND_DATA_XFER_MODE_DIO_GR3& = 14500
Global Const ND_DATA_XFER_MODE_DIO_GR4& = 14600
Global Const ND_DATA_XFER_MODE_DIO_GR5& = 14700
Global Const ND_DATA_XFER_MODE_DIO_GR6& = 14800
Global Const ND_DATA_XFER_MODE_DIO_GR7& = 14900
Global Const ND_DATA_XFER_MODE_DIO_GR8& = 15000

Global Const ND_DATA_XFER_MODE_GPCTR0& = 15100
Global Const ND_DATA_XFER_MODE_GPCTR1& = 15200
Global Const ND_DATA_XFER_MODE_GPCTR2& = 15110
Global Const ND_DATA_XFER_MODE_GPCTR3& = 15120
Global Const ND_DATA_XFER_MODE_GPCTR4& = 15130
Global Const ND_DATA_XFER_MODE_GPCTR5& = 15140
Global Const ND_DATA_XFER_MODE_GPCTR6& = 15150
Global Const ND_DATA_XFER_MODE_GPCTR7& = 15160
Global Const ND_DATA_XFER_MODE_GPCTR8& = 15165
Global Const ND_DATA_XFER_MODE_GPCTR9& = 15170
Global Const ND_DATA_XFER_MODE_GPCTR10& = 15175
Global Const ND_DATA_XFER_MODE_GPCTR11& = 15180

Global Const ND_DC& = 15250
Global Const ND_DDS_BUFFER_SIZE& = 15255
Global Const ND_DEVICE_NAME& = 15260
Global Const ND_DEVICE_POWER& = 15270
Global Const ND_DEVICE_SERIAL_NUMBER& = 15280
Global Const ND_DEVICE_STATE_DURING_SUSPEND_MODE& = 15290
Global Const ND_DEVICE_TYPE_CODE& = 15300
Global Const ND_DIGITAL_FILTER& = 15350
Global Const ND_DIGITAL_RESTART& = 15375
```

```
Global Const ND_DIO128_GET_PORT_THRESHOLD& = 41200
Global Const ND_DIO128_SELECT_INPUT_PORT& = 41100
Global Const ND_DIO128_SET_PORT_THRESHOLD& = 41300
Global Const ND_DISABLED& = 15400
Global Const ND_DISARM& = 15450
Global Const ND_DIVIDE_DOWN_SAMPLING_SUPPORTED& = 15475
Global Const ND_DMA_A_LEVEL& = 15500
Global Const ND_DMA_B_LEVEL& = 15600
Global Const ND_DMA_C_LEVEL& = 15700
Global Const ND_DONE& = 15800
Global Const ND_DONT_CARE& = 15900
Global Const ND_DONT_KNOW& = 15950

Global Const ND_EDGE_SENSITIVE& = 16000
Global Const ND_ENABLED& = 16050
Global Const ND_END& = 16055
Global Const ND_EXTERNAL& = 16060
Global Const ND_EXTERNAL_CALIBRATE& = 16100

Global Const ND_FACTORY_CALIBRATION_EQUIP& = 16210
Global Const ND_FACTORY_EEPROM_AREA& = 16220
Global Const ND_FIFO_EMPTY& = 16230
Global Const ND_FIFO_HALF_FULL_OR_LESS& = 16240
Global Const ND_FIFO_HALF_FULL_OR_LESS_UNTIL_FULL& = 16245
Global Const ND_FIFO_NOT_FULL& = 16250
Global Const ND_FIFO_TRANSFER_COUNT& = 16260
Global Const ND_FILTER_CORRECTION_FREQ& = 16300
Global Const ND_FOREGROUND& = 16350
Global Const ND_FREQ_OUT& = 16400
Global Const ND_FSK& = 16500
Global Const ND_EDGE_BASED_FSK& = 16500

Global Const ND_GATE& = 17100
Global Const ND_GATE_POLARITY& = 17200

Global Const ND_GPCTR0_GATE& = 17300
Global Const ND_GPCTR0_OUTPUT& = 17400
Global Const ND_GPCTR0_SOURCE& = 17500

Global Const ND_GPCTR1_GATE& = 17600
Global Const ND_GPCTR1_OUTPUT& = 17700
Global Const ND_GPCTR1_SOURCE& = 17800

Global Const ND_GPCTR2_GATE& = 17320
Global Const ND_GPCTR2_OUTPUT& = 17420
Global Const ND_GPCTR2_SOURCE& = 17520

Global Const ND_GPCTR3_GATE& = 17330
Global Const ND_GPCTR3_OUTPUT& = 17430
Global Const ND_GPCTR3_SOURCE& = 17530

Global Const ND_GPCTR4_GATE& = 17340
Global Const ND_GPCTR4_OUTPUT& = 17440
Global Const ND_GPCTR4_SOURCE& = 17540

Global Const ND_GPCTR5_GATE& = 17350
Global Const ND_GPCTR5_OUTPUT& = 17450
Global Const ND_GPCTR5_SOURCE& = 17550

Global Const ND_GPCTR6_GATE& = 17360
Global Const ND_GPCTR6_OUTPUT& = 17460
Global Const ND_GPCTR6_SOURCE& = 17660

Global Const ND_GPCTR7_GATE& = 17370
Global Const ND_GPCTR7_OUTPUT& = 17470
Global Const ND_GPCTR7_SOURCE& = 17570
```

```
Global Const ND_GROUND_DAC_REFERENCE& = 17900

Global Const ND_HARDWARE& = 18000
Global Const ND_HI_RES_SAMPLING& = 18020
Global Const ND_HIGH& = 18050
Global Const ND_HIGH_HYSTERESIS& = 18080
Global Const ND_HIGH_TO_LOW& = 18100
Global Const ND_HW_ANALOG_TRIGGER& = 18900

Global Const ND_IMPEDANCE& = 19000
Global Const ND_INACTIVE& = 19010
Global Const ND_INITIAL_COUNT& = 19100
Global Const ND_INIT_PLUGPLAY_DEVICES& = 19110
Global Const ND_INSIDE_REGION& = 19150
Global Const ND_INTERNAL& = 19160
Global Const ND_INTERNAL_100_KHZ& = 19200
Global Const ND_INTERNAL_10_MHZ& = 19300
Global Const ND_INTERNAL_1250_KHZ& = 19320
Global Const ND_INTERNAL_20_MHZ& = 19400
Global Const ND_INTERNAL_25_MHZ& = 19410
Global Const ND_INTERNAL_2500_KHZ& = 19420
Global Const ND_INTERNAL_5_MHZ& = 19450
Global Const ND_INTERNAL_7160_KHZ& = 19460
Global Const ND_INTERNAL_TIMER& = 19500
Global Const ND_INTERRUPTS& = 19600
Global Const ND_INTERRUPT_A_LEVEL& = 19700
Global Const ND_INTERRUPT_B_LEVEL& = 19800
Global Const ND_INTERRUPT_TRIGGER_MODE& = 19850
Global Const ND_IN_CHANNEL_CLOCK_TIMEBASE& = 19900
Global Const ND_IN_CHANNEL_CLOCK_TB_POL& = 20000
Global Const ND_IN_CONVERT& = 20100
Global Const ND_IN_CONVERT_POL& = 20200
Global Const ND_IN_DATA_FIFO_SIZE& = 20250
Global Const ND_IN_EXTERNAL_GATE& = 20300
Global Const ND_IN_EXTERNAL_GATE_POL& = 20400
Global Const ND_IN_SCAN_CLOCK_TIMEBASE& = 20500
Global Const ND_IN_SCAN_CLOCK_TB_POL& = 20600
Global Const ND_IN_SCAN_IN_PROG& = 20650
Global Const ND_IN_SCAN_START& = 20700
Global Const ND_IN_SCAN_START_POL& = 20800
Global Const ND_IN_START_TRIGGER& = 20900
Global Const ND_IN_START_TRIGGER_POL& = 21000
Global Const ND_IN_STOP_TRIGGER& = 21100
Global Const ND_IN_STOP_TRIGGER_POL& = 21200
Global Const ND_INT_AI_GND& = 21210
Global Const ND_INT_AO_CH_0& = 21230
Global Const ND_INT_AO_CH_0_VS_REF_5V& = 21235
Global Const ND_INT_AO_CH_1& = 21240
Global Const ND_INT_AO_CH_1_VS_AO_CH_0& = 21245
Global Const ND_INT_AO_CH_1_VS_REF_5V& = 21250
Global Const ND_INT_AO_CH_2& = 21220
Global Const ND_INT_AO_CH_3& = 21221
Global Const ND_INT_AO_CH_4& = 21222
Global Const ND_INT_AO_CH_5& = 21223
Global Const ND_INT_AO_CH_6& = 21224
Global Const ND_INT_AO_CH_7& = 21225
Global Const ND_INT_AO_GND& = 21260
Global Const ND_INT_AO_GND_VS_AI_GND& = 21265
Global Const ND_INT_CM_REF_5V& = 21270
Global Const ND_INT_DEV_TEMP& = 21280
Global Const ND_INT_REF_5V& = 21290
```

```
Global Const ND_INT_REF_EXTERN& = 21296
Global Const ND_INT_CAL_BUS& = 21295
Global Const ND_INT_MUX_BUS& = 21305

Global Const ND_INT_AI_GND_AMP_0& = 21211
Global Const ND_INT_AI_GND_AMP_1& = 21212
Global Const ND_INT_AI_GND_AMP_2& = 21213
Global Const ND_INT_AI_GND_AMP_3& = 21214
Global Const ND_INT_AO_CH_0_AMP_0& = 21231
Global Const ND_INT_AO_CH_0_AMP_1& = 21232
Global Const ND_INT_AO_CH_0_AMP_2& = 21233
Global Const ND_INT_AO_CH_0_AMP_3& = 21234
Global Const ND_INT_AO_CH_1_AMP_0& = 21241
Global Const ND_INT_AO_CH_1_AMP_1& = 21242
Global Const ND_INT_AO_CH_1_AMP_2& = 21243
Global Const ND_INT_AO_CH_1_AMP_3& = 21244
Global Const ND_INT_AO_CH_0_VS_REF_AMP_0& = 21236
Global Const ND_INT_AO_CH_0_VS_REF_AMP_1& = 21237
Global Const ND_INT_AO_CH_0_VS_REF_AMP_2& = 21238
Global Const ND_INT_AO_CH_0_VS_REF_AMP_3& = 21239
Global Const ND_INT_AO_CH_1_VS_REF_AMP_0& = 21251
Global Const ND_INT_AO_CH_1_VS_REF_AMP_1& = 21252
Global Const ND_INT_AO_CH_1_VS_REF_AMP_2& = 21253
Global Const ND_INT_AO_CH_1_VS_REF_AMP_3& = 21254
Global Const ND_INT_AO_GND_VS_AI_GND_AMP_0& = 21266
Global Const ND_INT_AO_GND_VS_AI_GND_AMP_1& = 21267
Global Const ND_INT_AO_GND_VS_AI_GND_AMP_2& = 21268
Global Const ND_INT_AO_GND_VS_AI_GND_AMP_3& = 21269
Global Const ND_INT_CM_REF_AMP_0& = 21271
Global Const ND_INT_CM_REF_AMP_1& = 21272
Global Const ND_INT_CM_REF_AMP_2& = 21273
Global Const ND_INT_CM_REF_AMP_3& = 21274
Global Const ND_INT_REF_AMP_0& = 21291
Global Const ND_INT_REF_AMP_1& = 21292
Global Const ND_INT_REF_AMP_2& = 21293
Global Const ND_INT_REF_AMP_3& = 21294

Global Const ND_INTERRUPT_EVERY_SAMPLE& = 11700
Global Const ND_INTERRUPT_HALF_FIFO& = 11800
Global Const ND_IO_CONNECTOR& = 21300

Global Const ND_LEVEL_SENSITIVE& = 24000
Global Const ND_LINK_COMPLETE_INTERRUPTS& = 24010
Global Const ND_LOW& = 24050
Global Const ND_LOW_HYSTERESIS& = 24080
Global Const ND_LOW_TO_HIGH& = 24100
Global Const ND_LPT_DEVICE_MODE& = 24200

Global Const ND_MARKER& = 24500
Global Const ND_MARKER_QUANTUM& = 24550
Global Const ND_MAX_ARB_SEQUENCE_LENGTH& = 24600
Global Const ND_MAX_FUNC_SEQUENCE_LENGTH& = 24610
Global Const ND_MAX_LOOP_COUNT& = 24620
Global Const ND_MAX_NUM_WAVEFORMS& = 24630
Global Const ND_MAX_SAMPLE_RATE& = 24640
Global Const ND_MAX_WFM_SIZE& = 24650
Global Const ND_MEMORY_TRANSFER_WIDTH& = 24700
Global Const ND_MIN_SAMPLE_RATE& = 24800
Global Const ND_MIN_WFM_SIZE& = 24810

Global Const ND_NEGATIVE& = 26100
Global Const ND_NEW& = 26190
Global Const ND_NI_DAQ_SW_AREA& = 26195
Global Const ND_NO& = 26200
Global Const ND_NO_STRAIN_GAUGE& = 26225
Global Const ND_NO_TRACK_AND_HOLD& = 26250
```

```
Global Const ND_NONE& = 26300
Global Const ND_NOT_APPLICABLE& = 26400
Global Const ND_NUMBER_DIG_PORTS& = 26500

Global Const ND_OFF& = 27010
Global Const ND_OFFSET& = 27020
Global Const ND_ON& = 27050
Global Const ND_OTHER& = 27060
Global Const ND_OTHER_GPCTR_OUTPUT& = 27300
Global Const ND_OTHER_GPCTR_TC& = 27400
Global Const ND_OUT_DATA_FIFO_SIZE& = 27070
Global Const ND_OUT_EXTERNAL_GATE& = 27080
Global Const ND_OUT_EXTERNAL_GATE_POL& = 27082
Global Const ND_OUT_START_TRIGGER& = 27100
Global Const ND_OUT_START_TRIGGER_POL& = 27102
Global Const ND_OUT_UPDATE& = 27200
Global Const ND_OUT_UPDATE_POL& = 27202
Global Const ND_OUT_UPDATE_CLOCK_TIMEBASE& = 27210
Global Const ND_OUT_UPDATE_CLOCK_TB_POL& = 27212
Global Const ND_OUTPUT_ENABLE& = 27220
Global Const ND_OUTPUT_MODE& = 27230
Global Const ND_OUTPUT_POLARITY& = 27240
Global Const ND_OUTPUT_STATE& = 27250
Global Const ND_OUTPUT_TYPE& = 40000

Global Const ND_DIGITAL_PATTERN_GENERATION& = 28030
Global Const ND_PAUSE& = 28040
Global Const ND_PAUSE_ON_HIGH& = 28045
Global Const ND_PAUSE_ON_LOW& = 28050
Global Const ND_PFI_0& = 28100
Global Const ND_PFI_1& = 28200
Global Const ND_PFI_2& = 28300
Global Const ND_PFI_3& = 28400
Global Const ND_PFI_4& = 28500
Global Const ND_PFI_5& = 28600
Global Const ND_PFI_6& = 28700
Global Const ND_PFI_7& = 28800
Global Const ND_PFI_8& = 28900
Global Const ND_PFI_9& = 29000
Global Const ND_PFI_10& = 50280
Global Const ND_PFI_11& = 50290
Global Const ND_PFI_12& = 50300
Global Const ND_PFI_13& = 50310
Global Const ND_PFI_14& = 50320
Global Const ND_PFI_15& = 50330
Global Const ND_PFI_16& = 50340
Global Const ND_PFI_17& = 50350
Global Const ND_PFI_18& = 50360
Global Const ND_PFI_19& = 50370
Global Const ND_PFI_20& = 50380
Global Const ND_PFI_21& = 50390
Global Const ND_PFI_22& = 50400
Global Const ND_PFI_23& = 50410
Global Const ND_PFI_24& = 50420
Global Const ND_PFI_25& = 50430
Global Const ND_PFI_26& = 50440
Global Const ND_PFI_27& = 50450
Global Const ND_PFI_28& = 50460
Global Const ND_PFI_29& = 50470
Global Const ND_PFI_30& = 50480
Global Const ND_PFI_31& = 50490
Global Const ND_PFI_32& = 50500
Global Const ND_PFI_33& = 50510
Global Const ND_PFI_34& = 50520
Global Const ND_PFI_35& = 50530
Global Const ND_PFI_36& = 50540
```

```
Global Const ND_PFI_37& = 50550
Global Const ND_PFI_38& = 50560
Global Const ND_PFI_39& = 50570

Global Const ND_PLL_REF_FREQ& = 29010
Global Const ND_PLL_REF_SOURCE& = 29020
Global Const ND_PRE_ARM& = 29050
Global Const ND_POSITIVE& = 29100
Global Const ND_PREPARE& = 29200
Global Const ND_PROGRAM& = 29300
Global Const ND_PULSE& = 29350
Global Const ND_PULSE_SOURCE& = 29500
Global Const ND_PULSE_TRAIN_GNR& = 29600
Global Const ND_PXI_BACKPLANE_CLOCK& = 29900

Global Const ND_REGLITCH& = 31000
Global Const ND_RESERVED& = 31100
Global Const ND_RESET& = 31200
Global Const ND_RESUME& = 31250
Global Const ND_RETRIG_PULSE_GNR& = 31300
Global Const ND_REVISION& = 31350
Global Const ND_RTSI_0& = 31400
Global Const ND_RTSI_1& = 31500
Global Const ND_RTSI_2& = 31600
Global Const ND_RTSI_3& = 31700
Global Const ND_RTSI_4& = 31800
Global Const ND_RTSI_5& = 31900
Global Const ND_RTSI_6& = 32000
Global Const ND_RTSI_CLOCK& = 32100

Global Const ND_SCANCLK& = 32400
Global Const ND_SCANCLK_LINE& = 32420
Global Const ND_SC_2040_MODE& = 32500
Global Const ND_SC_2043_MODE& = 32600
Global Const ND_SELF_CALIBRATE& = 32700
Global Const ND_SET_DEFAULT_LOAD_AREA& = 32800
Global Const ND_RESTORE_FACTORY_CALIBRATION& = 32810
Global Const ND_SET_POWERUP_STATE& = 42100
Global Const ND_SIMPLE_EVENT_CNT& = 33100
Global Const ND_SINGLE& = 33150
Global Const ND_SINGLE_PERIOD_MSR& = 33200
Global Const ND_SINGLE_PULSE_GNR& = 33300
Global Const ND_SINGLE_PULSE_WIDTH_MSR& = 33400
Global Const ND_SINGLE_TRIG_PULSE_GNR& = 33500
Global Const ND_SOURCE& = 33700
Global Const ND_SOURCE_POLARITY& = 33800
Global Const ND_STABLE_10_MHZ& = 33810
Global Const ND_STEPPED& = 33825
Global Const ND_STRAIN_GAUGE& = 33850
Global Const ND_STRAIN_GAUGE_EX0& = 33875
Global Const ND_SUB_REVISION& = 33900
Global Const ND_SYNC_DUTY_CYCLE_HIGH& = 33930
Global Const ND_SYNC_OUT& = 33970

Global Const ND_TC_REACHED& = 34100
Global Const ND_THE_AI_CHANNEL& = 34400
Global Const ND_TOGGLE& = 34700
Global Const ND_TOGGLE_GATE& = 34800
Global Const ND_TRACK_AND_HOLD& = 34850
Global Const ND_TRIG_PULSE_WIDTH_MSR& = 34900
Global Const ND_TRIGGER_SOURCE& = 34930
Global Const ND_TRIGGER_MODE& = 34970

Global Const ND_UI2_TC& = 35100
Global Const ND_UP_DOWN& = 35150
Global Const ND_UP_TO_1_DMA_CHANNEL& = 35200
```

```
Global Const ND_UP_TO_2_DMA_CHANNELS& = 35300
Global Const ND_USE_CAL_CHAN& = 36000
Global Const ND_USE_AUX_CHAN& = 36100
Global Const ND_USER_EEPROM_AREA& = 37000
Global Const ND_USER_EEPROM_AREA_2& = 37010
Global Const ND_USER_EEPROM_AREA_3& = 37020
Global Const ND_USER_EEPROM_AREA_4& = 37030
Global Const ND_USER_EEPROM_AREA_5& = 37040

Global Const ND_DSA_RTSI_CLOCK_AD& = 44000
Global Const ND_DSA_RTSI_CLOCK_DA& = 44010
Global Const ND_DSA_OUTPUT_TRIGGER& = 44020
Global Const ND_DSA_INPUT_TRIGGER& = 44030
Global Const ND_DSA_SHARC_TRIGGER& = 44040
Global Const ND_DSA_ANALOG_TRIGGER& = 44050
Global Const ND_DSA_HOST_TRIGGER& = 44060
Global Const ND_DSA_EXTERNAL_DIGITAL_TRIGGER& = 44070

Global Const ND_VOLTAGE_OUTPUT& = 40100
Global Const ND_VOLTAGE_REFERENCE& = 38000

Global Const ND_VXI_SC% = &H2000
Global Const ND_PXI_SC% = &H2010
Global Const ND_VXIMIO_SET_ALLOCATE_MODE& = 43100
Global Const ND_VXIMIO_USE_ONBOARD_MEMORY_AI& = 43500
Global Const ND_VXIMIO_USE_ONBOARD_MEMORY_AO& = 43600
Global Const ND_VXIMIO_USE_ONBOARD_MEMORY_GPCTR& = 43700
Global Const ND_VXIMIO_USE_PC_MEMORY_AI& = 43200
Global Const ND_VXIMIO_USE_PC_MEMORY_AO& = 43300
Global Const ND_VXIMIO_USE_PC_MEMORY_GPCTR& = 43400

Global Const ND_WFM_QUANTUM& = 45000

Global Const ND_YES& = 39100
Global Const ND_3V_LEVEL& = 43450

Global Const ND_WRITE_MARK& = 50000
Global Const ND_READ_MARK& = 50010
Global Const ND_BUFFER_START& = 50020
Global Const ND_TRIGGER_POINT& = 50025
Global Const ND_BUFFER_MODE& = 50030
Global Const ND_DOUBLE& = 50050
Global Const ND_QUADRATURE_ENCODER_X1& = 50070
Global Const ND_QUADRATURE_ENCODER_X2& = 50080
Global Const ND_QUADRATURE_ENCODER_X4& = 50090
Global Const ND_TWO_PULSE_COUNTING& = 50100
Global Const ND_LINE_FILTER& = 50110
Global Const ND_SYNCHRONIZATION& = 50120
Global Const ND_5_MICROSECONDS& = 50130
Global Const ND_1_MICROSECOND& = 50140
Global Const ND_500_NANOSECONDS& = 50150
Global Const ND_100_NANOSECONDS& = 50160
Global Const ND_1_MILLISECOND& = 50170
Global Const ND_10_MILLISECONDS& = 50180
Global Const ND_100_MILLISECONDS& = 50190


Global Const ND_OTHER_GPCTR_SOURCE& = 50580
Global Const ND_OTHER_GPCTR_GATE& = 50590
Global Const ND_AUX_LINE& = 50600
Global Const ND_AUX_LINE_POLARITY& = 50610
Global Const ND_TWO_SIGNAL_EDGE_SEPARATION_MSR& = 50630
Global Const ND_BUFFERED_TWO_SIGNAL_EDGE_SEPARATION_MSR& = 50640
Global Const ND_SWITCH_CYCLE& = 50650
Global Const ND_INTERNAL_MAX_TIMEBASE& = 50660
Global Const ND_PRESCALE_VALUE& = 50670
```

```
Global Const ND_MAX_PRESCALE& = 50690
Global Const ND_INTERNAL_LINE_0& = 50710
Global Const ND_INTERNAL_LINE_1& = 50720
Global Const ND_INTERNAL_LINE_2& = 50730
Global Const ND_INTERNAL_LINE_3& = 50740
Global Const ND_INTERNAL_LINE_4& = 50750
Global Const ND_INTERNAL_LINE_5& = 50760
Global Const ND_INTERNAL_LINE_6& = 50770
Global Const ND_INTERNAL_LINE_7& = 50780
Global Const ND_INTERNAL_LINE_8& = 50790
Global Const ND_INTERNAL_LINE_9& = 50800
Global Const ND_INTERNAL_LINE_10& = 50810
Global Const ND_INTERNAL_LINE_11& = 50820
Global Const ND_INTERNAL_LINE_12& = 50830
Global Const ND_INTERNAL_LINE_13& = 50840
Global Const ND_INTERNAL_LINE_14& = 50850
Global Const ND_INTERNAL_LINE_15& = 50860
Global Const ND_INTERNAL_LINE_16& = 50862
Global Const ND_INTERNAL_LINE_17& = 50864
Global Const ND_INTERNAL_LINE_18& = 50866
Global Const ND_INTERNAL_LINE_19& = 50868
Global Const ND_INTERNAL_LINE_20& = 50870
Global Const ND_INTERNAL_LINE_21& = 50872
Global Const ND_INTERNAL_LINE_22& = 50874
Global Const ND_INTERNAL_LINE_23& = 50876



Global Const ND_START_TRIGGER& = 51150
Global Const ND_START_TRIGGER_POLARITY& = 51151


Global Const ND_COUNTING_SYNCHRONOUS& = 51200
Global Const ND_SYNCHRONOUS& = 51210
Global Const ND_ASYNCHRONOUS& = 51220
Global Const ND_CONFIGURABLE_FILTER& = 51230
Global Const ND_ENCODER_TYPE& = 51240
Global Const ND_Z_INDEX_ACTIVE& = 51250
Global Const ND_Z_INDEX_VALUE& = 51260
Global Const ND_SNAPSHOT& = 51270
Global Const ND_POSITION_MSR& = 51280
Global Const ND_BUFFERED_POSITION_MSR& = 51290
Global Const ND_SAVED_COUNT& = 51300
Global Const ND_READ_MARK_H_SNAPSHOT& = 51310
Global Const ND_READ_MARK_L_SNAPSHOT& = 51320
Global Const ND_WRITE_MARK_H_SNAPSHOT& = 51330
Global Const ND_WRITE_MARK_L_SNAPSHOT& = 51340
Global Const ND_BACKLOG_H_SNAPSHOT& = 51350
Global Const ND_BACKLOG_L_SNAPSHOT& = 51360
Global Const ND_ARMED_SNAPSHOT& = 51370
Global Const ND_EDGE_GATED_FSK& = 51371
Global Const ND_SIMPLE_GATED_EVENT_CNT& = 51372

Global Const ND_VIDEO_TYPE& = 51380
Global Const ND_PAL_B& = 51390
Global Const ND_PAL_G& = 51400
Global Const ND_PAL_H& = 51410
Global Const ND_PAL_I& = 51420
Global Const ND_PAL_D& = 51430
Global Const ND_PAL_N& = 51440
Global Const ND_PAL_M& = 51450
Global Const ND_NTSC_M& = 51460
Global Const ND_COUNTER_TYPE& = 51470
```

```
Global Const ND_NI_TIO& = 51480
Global Const ND_AM9513& = 51490
Global Const ND_STC& = 51500
Global Const ND_8253& = 51510
Global Const ND_A_HIGH_B_HIGH& = 51520
Global Const ND_A_HIGH_B_LOW& = 51530
Global Const ND_A_LOW_B_HIGH& = 51540
Global Const ND_A_LOW_B_LOW& = 51550
Global Const ND_Z_INDEX_RELOAD_PHASE& = 51560
Global Const ND_UPDOWN_LINE& = 51570
Global Const ND_DEFAULT_PFI_LINE& = 51580
Global Const ND_BUFFER_SIZE& = 51590
Global Const ND_ELEMENT_SIZE& = 51600
Global Const ND_NUMBER_GP_COUNTERS& = 51610
Global Const ND_BUFFERED_TIME_STAMPING& = 51620
Global Const ND_TIME_0_DATA_32& = 51630
Global Const ND_TIME_8_DATA_24& = 51640
Global Const ND_TIME_16_DATA_16& = 51650
Global Const ND_TIME_24_DATA_8& = 51660
Global Const ND_TIME_32_DATA_32& = 51670
Global Const ND_TIME_48_DATA_16& = 51680
Global Const ND_ABSOLUTE& = 51690
Global Const ND_RELATIVE& = 51700
Global Const ND_TIME_DATA_SIZE& = 51710
Global Const ND_TIME_FORMAT& = 51720
Global Const ND_HALT_ON_OVERFLOW& = 51730
Global Const ND_OVERLAY_RTSI_ON_PFI_LINES& = 51740
Global Const ND_STOP_TRIGGER& = 51750
Global Const ND_TS_INPUT_MODE& = 51760
Global Const ND_BOTH_EDGES& = 51770

Global Const ND_CLOCK_0& = 51780
Global Const ND_CLOCK_1& = 51790
Global Const ND_CLOCK_2& = 51800
Global Const ND_CLOCK_3& = 51810
Global Const ND_SYNCHRONIZATION_LINE& = 51820
Global Const ND_TRANSFER_METHOD& = 51830
Global Const ND_SECONDS& = 51840
Global Const ND_PRECISION& = 51850
Global Const ND_NANO_SECONDS& = 51860
Global Const ND_SYNCHRONIZATION_METHOD& = 51870
Global Const ND_PULSE_PER_SECOND& = 51880
Global Const ND_IRIG_B& = 51890
Global Const ND_SIMPLE_TIME_MSR& = 51900
Global Const ND_SINGLE_TIME_MSR& = 51910
Global Const ND_BUFFERED_TIME_MSR& = 51920
Global Const ND_DMA& = 51930
```

NIDAQERR.INC Module

```
'*********************************************************************'
'* nidaqerr.inc                                    *'
'*   header file for platform-independent ni-daq errors/warnings   *'
'*                                                 *'
'* NOTE:                                           *'
'*   You should use symbols defined here in your programs;  do not  *'
'*   use the numerical values.                     *'
'*                                                 *'
'*   Warnings are returned as positive numbers. For example       *'
'*   overWriteError may be returned as a warning and its value    *'
'*   would be -(overWriteError).                   *'
'*                                                 *'
'* THIS FILE IS AUTOMATICALLY GENERATED FROM A DATABASE: DO NOT EDIT *'
'*                                                 *'
'*********************************************************************'

Global Const noError = 0
```

```
Global Const syntaxError = -10001              'An error was detected in the input string; the arrangement or ordering ...
Global Const semanticsError = -10002           'An error was detected in the input string; the syntax of the string is ...
Global Const invalidValueError = -10003        'The value of a numeric parameter is invalid.
Global Const valueConflictError = -10004       'The value of a numeric parameter is inconsistent with another one, and ...
Global Const badDeviceError = -10005           'The device is invalid.
Global Const badLineError = -10006             'The line is invalid.
Global Const badChanError = -10007             'A channel, port, or counter is out of range for the device type or device ...
Global Const badGroupError = -10008            'The group is invalid.
Global Const badCounterError = -10009          'The counter is invalid.
Global Const badCountError = -10010            'The count is too small or too large for the specified counter, or the ...
Global Const badIntervalError = -10011         'The analog input scan rate is too fast for the number of channels and ...
Global Const badRangeError = -10012            'The analog input or analog output voltage or current range is invalid ...
Global Const badErrorCodeError = -10013        'The driver returned an unrecognized or unlisted error code.
Global Const groupTooLargeError = -10014       'The group size is too large for the board.
Global Const badTimeLimitError = -10015        'The time limit is invalid.
Global Const badReadCountError = -10016        'The read count is invalid.
Global Const badReadModeError = -10017         'The read mode is invalid.
Global Const badReadOffsetError = -10018       'The offset is unreachable.
Global Const badClkFrequencyError = -10019     'The frequency is invalid.
Global Const badTimebaseError = -10020         'The timebase is invalid.
Global Const badLimitsError = -10021           'The limits are beyond the range of the board.
Global Const badWriteCountError = -10022       'Your data array contains an incomplete update, or you are trying to write ...
Global Const badWriteModeError = -10023        'The write mode is out of range or is disallowed.
Global Const badWriteOffsetError = -10024      'Adding the write offset to the write mark places the write mark outside ...
Global Const limitsOutOfRangeError = -10025    'The requested input limits exceed the board's capability or configuration. ...
Global Const badBufferSpecificationError = -10026   'The requested number of buffers or the buffer size is not allowed. For ...
Global Const badDAQEventError = -10027         'For DAQEvents 0 and 1 general value A must be greater than 0 and less ...
Global Const badFilterCutoffError = -10028     'The cutoff frequency specified is not valid for this device.
Global Const obsoleteFunctionError = -10029    'The function you are calling is no longer supported in this version of ...
Global Const badBaudRateError = -10030         'The specified baud rate for communicating with the serial port is not ...
Global Const badChassisIDError = -10031        'The specified baud rate for communicating with the serial port is not ...
Global Const badModuleSlotError = -10032       'The SCXI module slot that was specified is invalid or corresponds to an ...
Global Const invalidWinHandleError = -10033    'The window handle passed to the function is invalid.
Global Const noSuchMessageError = -10034       'No configured message matches the one you tried to delete.
Global Const irrelevantAttributeError = -10035 'The specified attribute is not relevant.
Global Const badYearError = -10036             'The specified year is invalid.
Global Const badMonthError = -10037            'The specified month is invalid.
Global Const badDayError = -10038              'The specified day is invalid.
Global Const stringTooLongError = -10039       'The specified input string is too long.  For instance, DAQScope 5102 devices ...
Global Const badGroupSizeError = -10040        'The group size is invalid.
Global Const badTaskIDError = -10041           'The specified task ID is invalid. For instance, you may have connected ...
Global Const inappropriateControlCodeError = -10042   'The specified control code is inappropriate for the current configuration ...
Global Const badDivisorError = -10043          'The specified divisor is invalid.
Global Const badPolarityError = -10044         'The specified polarity is invalid.
Global Const badInputModeError = -10045        'The specified input mode is invalid.
Global Const badExcitationError = -10046       'The excitation value specified is not valid for this device.
```

```
Global Const badConnectionTypeError = -10047     'The excitation value specified is not valid for this device.
Global Const badExcitationTypeError = -10048     'The excitation type specified is not valid for this device.
Global Const badChanListError = -10050     'There is more than one channel name in the channel list that corresponds ...
Global Const badTrigSkipCountError = -10079     'The trigger skip count is invalid.
Global Const badGainError = -10080     'The gain or gain adjust is invalid.
Global Const badPretrigCountError = -10081     'The pretrigger sample count is invalid.
Global Const badPosttrigCountError = -10082     'The posttrigger sample count is invalid.
Global Const badTrigModeError = -10083     'The trigger mode is invalid.
Global Const badTrigCountError = -10084     'The trigger count is invalid.
Global Const badTrigRangeError = -10085     'The trigger range or trigger hysteresis window is invalid.
Global Const badExtRefError = -10086     'The external reference  is invalid.
Global Const badTrigTypeError = -10087     'The trigger type is invalid.
Global Const badTrigLevelError = -10088     'The trigger level is invalid.
Global Const badTotalCountError = -10089     'The total count is inconsistent with the buffer size and pretrigger scan ...
Global Const badRPGError = -10090     'The individual range, polarity, and gain settings are valid but the combination ...
Global Const badIterationsError = -10091     'You have attempted to use an invalid setting for the iterations parameter. ...
Global Const lowScanIntervalError = -10092     'Some devices require a time gap between the last sample in a scan and ...
Global Const fifoModeError = -10093     'FIFO mode waveform generation cannot be used because at least one condition ...
Global Const badCalDACconstError = -10094     'The calDAC constant passed to the function is invalid.
Global Const badCalStimulusError = -10095     'The calibration stimulus passed to the function is invalid.
Global Const badCalibrationConstantError = -10096     'The specified calibration constant is invalid.
Global Const badCalOpError = -10097     'The specified calibration operation is invalid.
Global Const badCalConstAreaError = -10098     'The specified calibration constant area is invalid.  For instance, the ...
Global Const badPortWidthError = -10100     'The requested digital port width is not a multiple of the hardware port ...
Global Const gpctrBadApplicationError = -10120     'Invalid application used.
Global Const gpctrBadCtrNumberError = -10121     'Invalid counterNumber used.
Global Const gpctrBadParamValueError = -10122     'Invalid paramValue used.
Global Const gpctrBadParamIDError = -10123     'Invalid paramID used.
Global Const gpctrBadEntityIDError = -10124     'Invalid entityID used.
Global Const gpctrBadActionError = -10125     'Invalid action used.
Global Const gpctrSourceSelectError = -10126     'Invalid source selected.
Global Const badCountDirError = -10127     'The specified counter does not support the specified count direction.
Global Const badGateOptionError = -10128     'The specified gating option is invalid.
Global Const badGateModeError = -10129     'The specified gate mode is invalid.
Global Const badGateSourceError = -10130     'The specified gate source is invalid.
Global Const badGateSignalError = -10131     'The specified gate signal is invalid.
Global Const badSourceEdgeError = -10132     'The specified source edge is invalid.
Global Const badOutputTypeError = -10133     'The specified output type is invalid.
Global Const badOutputPolarityError = -10134     'The specified output polarity is invalid.
Global Const badPulseModeError = -10135     'The specified pulse mode is invalid.
Global Const badDutyCycleError = -10136     'The specified duty cycle is invalid.
Global Const badPulsePeriodError = -10137     'The specified pulse period is invalid.
Global Const badPulseDelayError = -10138     'The specified pulse delay is invalid.
Global Const badPulseWidthError = -10139     'The specified pulse width is invalid.
Global Const badFOUTportError = -10140     'The specified frequency output (FOUT or FREQ_OUT) port is invalid.
Global Const badAutoIncrementModeError = -10141     'The specified autoincrement mode is invalid.
Global Const badNotchFilterError = -10180     'The specified notch filter is invalid.
Global Const badMeasModeError = -10181     'The specified measurement mode is invalid.
Global Const EEPROMreadError = -10200     'Unable to read data from EEPROM.
Global Const EEPROMwriteError = -10201     'Unable to write data to EEPROM.
Global Const EEPROMwriteProtectionError = -10202     'You cannot write into this location or area of your EEPROM because it ...
Global Const EEPROMinvalidLocationError = -10203     'The specified EEPROM location is invalid.
Global Const EEPROMinvalidPasswordError = -10204     'The password for accessing the EEPROM is incorrect.
Global Const noDriverError = -10240     'The driver interface could not locate or open the driver..
```

```
Global Const oldDriverError = -10241                'One of the driver files or the configuration utility is out of date, or ...
Global Const functionNotFoundError = -10242         'The specified function is not located in the driver.
Global Const configFileError = -10243               'The driver could not locate or open the configuration file, or the format ...
Global Const deviceInitError = -10244               'The driver encountered a hardware-initialization error while attempting ...
Global Const osInitError = -10245                   'The driver encountered an operating-system error while attempting to perform ...
Global Const communicationsError = -10246           'The driver encountered an operating-system error while attempting to perform ...
Global Const cmosConfigError = -10247               'The CMOS configuration-memory for the device is empty or invalid, or the ...
Global Const dupAddressError = -10248               'The base addresses for two or more devices are the same; consequently, ...
Global Const intConfigError = -10249                'The interrupt configuration is incorrect given the capabilities of the ...
Global Const dupIntError = -10250                   'The interrupt levels for two or more devices are the same.
Global Const dmaConfigError = -10251                'The DMA configuration is incorrect given the capabilities of the computer/DMA ...
Global Const dupDMAError = -10252                   'The DMA channels for two or more devices are the same.
Global Const jumperlessBoardError = -10253          'Unable to find one or more jumperless boards you have configured using ...
Global Const DAQCardConfError = -10254              'Cannot configure the DAQCard because 1) the correct version of the card ...
Global Const remoteChassisDriverInitError = -10255  'There was an error in initializing the driver for Remote SCXI.
Global Const comPortOpenError = -10256              'There was an error in opening the specified COM port.
Global Const baseAddressError = -10257              'Bad base address specified in the configuration utility.
Global Const dmaChannel1Error = -10258              'Bad DMA channel 1 specified in the configuration utility or by the operating ...
Global Const dmaChannel2Error = -10259              'Bad DMA channel 2 specified in the configuration utility or by the operating ...
Global Const dmaChannel3Error = -10260              'Bad DMA channel 3 specified in the configuration utility or by the operating ...
Global Const userModeToKernelModeCallError = -10261 'The user mode code failed when calling the kernel mode code.
Global Const noConnectError = -10340                'No RTSI or PFI signal/line is connected, or the specified signal and the ...
Global Const badConnectError = -10341               'The RTSI or PFI signal/line cannot be connected as specified.
Global Const multConnectError = -10342              'The specified RTSI signal is already being driven by a RTSI line, or the ...
Global Const SCXIConfigError = -10343               'The specified SCXI configuration parameters are invalid, or the function ...
Global Const chassisSynchedError = -10344           'The Remote SCXI unit is not synchronized with the host. Reset the chassis ...
Global Const chassisMemAllocError = -10345          'The required amount of memory cannot be allocated on the Remote SCXI unit ...
Global Const badPacketError = -10346                'The packet received by the Remote SCXI unit is invalid. Check your serial ...
Global Const chassisCommunicationError = -10347     'There was an error in sending a packet to the remote chassis. Check your ...
Global Const waitingForReprogError = -10348         'The Remote SCXI unit is in reprogramming mode and is waiting for reprogramming ...
Global Const SCXIModuleTypeConflictError = -10349   'The module ID read from the SCXI module conflicts with the configured ...
Global Const CannotDetermineEntryModuleError = -10350 'Neither an SCXI entry module (i.e.: the SCXI module cabled to the measurement ...
Global Const DSPInitError = -10360                  'The DSP driver was unable to load the kernel for its operating system.
Global Const badScanListError = -10370              'The scan list is invalid; for example, you are mixing AMUX-64T channels ...
Global Const invalidSignalSrcError = -10380         'The specified signal source is invalid for the selected signal name.
Global Const invalidSignalNameError = -10381        'The specified signal name is invalid.
```

```
Global Const invalidSrcSpecError = -10382          'The specified source specification is invalid for the signal source or ...
Global Const invalidSignalDestError = -10383       'The specified signal destination is invalid.
Global Const userOwnedRsrcError = -10400           'The specified resource is owned by the user and cannot be accessed or ...
Global Const unknownDeviceError = -10401           'The specified device is not a National Instruments product, the driver ...
Global Const deviceNotFoundError = -10402          'The specified device is not a National Instruments product, the driver ...
Global Const deviceSupportError = -10403           'The specified device does not support the requested action (the driver ...
Global Const noLineAvailError = -10404             'No line is available.
Global Const noChanAvailError = -10405             'No channel is available.
Global Const noGroupAvailError = -10406            'No group is available.
Global Const lineBusyError = -10407                'The specified line is in use.
Global Const chanBusyError = -10408                'The specified channel is in use.
Global Const groupBusyError = -10409               'The specified group is in use.
Global Const relatedLCGBusyError = -10410          'A related line, channel, or group is in use; if the driver configures ...
Global Const counterBusyError = -10411             'The specified counter is in use.
Global Const noGroupAssignError = -10412           'No group is assigned, or the specified line or channel cannot be assigned ...
Global Const groupAssignError = -10413             'A group is already assigned, or the specified line or channel is already ...
Global Const reservedPinError = -10414             'The selected signal requires a pin that is reserved and configured only ...
Global Const externalMuxSupportError = -10415      'This function does not support your DAQ device when an external multiplexer ...
Global Const sysOwnedRsrcError = -10440            'The specified resource is owned by the driver and cannot be accessed or ...
Global Const memConfigError = -10441               'No memory is configured to support the current data-transfer mode, or ...
Global Const memDisabledError = -10442             'The specified memory is disabled or is unavailable given the current addressing ...
Global Const memAlignmentError = -10443            'The transfer buffer is not aligned properly for the current data-transfer ...
Global Const memFullError = -10444                 'No more system memory is available on the heap, or no more memory is available ...
Global Const memLockError = -10445                 'The transfer buffer cannot be locked into physical memory. On PC AT machines, ...
Global Const memPageError = -10446                 'The transfer buffer contains a page break; system resources may require ...
Global Const memPageLockError = -10447             'The operating environment is unable to grant a page lock.
Global Const stackMemError = -10448                'The operating environment is unable to grant a page lock.
Global Const cacheMemError = -10449                'A cache-related error occurred, or caching is not supported in the current ...
Global Const physicalMemError = -10450             'A hardware error occurred in physical memory, or no memory is located ...
Global Const virtualMemError = -10451              'The driver is unable to make the transfer buffer contiguous in virtual ...
Global Const noIntAvailError = -10452              'No interrupt level is available for use.
Global Const intInUseError = -10453                'The specified interrupt level is already in use by another device.
Global Const noDMACError = -10454                  'No DMA controller is available in the system.
Global Const noDMAAvailError = -10455              'No DMA channel is available for use.
Global Const DMAInUseError = -10456                'The specified DMA channel is already in use by another device.
Global Const badDMAGroupError = -10457             'DMA cannot be configured for the specified group because it is too small, ...
Global Const diskFullError = -10458                'The storage disk you specified is full.
Global Const DLLInterfaceError = -10459            'The NI-DAQ DLL could not be called due to an interface error.
Global Const interfaceInteractionError = -10460    'You have mixed VIs from the DAQ library and the _DAQ compatibility library ...
Global Const resourceReservedError = -10461        'The specified resource is unavailable because it has already been reserved ...
```

```
Global Const resourceNotReservedError = -10462        'The specified resource is unavailable because it has already been reserved ...
Global Const mdResourceAlreadyReservedError = -10463   'Another entity has already reserved the requested resource.
Global Const mdResourceReservedError = -10464         'Another entity has already reserved the requested resource.
Global Const mdResourceNotReservedError = -10465      'Attempting to lift a reservation off a resouce that previously had no ...
Global Const mdResourceAccessKeyError = -10466        'The requested operation cannot be performed because the key supplied is ...
Global Const mdResourceNotRegisteredError = -10467    'The resource requested is not registered with the minidriver.
Global Const muxMemFullError = -10480                 'The resource requested is not registered with the minidriver.
Global Const bufferNotInterleavedError = -10481       'You must provide a single buffer of interleaved data, and the channels ...
Global Const SCXIModuleNotSupportedError = -10540     'You must provide a single buffer of interleaved data, and the channels ...
Global Const TRIG1ResourceConflict = -10541           'CTRB1 will drive COUTB1, however CTRB1 will also drive TRIG1. This may ...
Global Const matrixTerminalBlockError = -10542        'This function requires that no Matrix terminal block is configured with ...
Global Const noMatrixTerminalBlockError = -10543      'This function requires that some matrix terminal block is configured with ...
Global Const invalidMatrixTerminalBlockError = -10544 'The type of matrix terminal block configured will not allow proper operation ...
Global Const invalidDSPHandleError = -10560           'The DSP handle input is not valid .
Global Const DSPDataPathBusyError = -10561            'Either DAQ or WFM can use a PC memory buffer, but not both at the same ...
Global Const noSetupError = -10600                    'No setup operation has been performed for the specified resources. Or, ...
Global Const multSetupError = -10601                  'No setup operation has been performed for the specified resources. Or, ...
Global Const noWriteError = -10602                    'No output data has been written into the transfer buffer.
Global Const groupWriteError = -10603                 'The output data associated with a group must be for a single channel or ...
Global Const activeWriteError = -10604                'Once data generation has started, only the transfer buffers originally ...
Global Const endWriteError = -10605                   'No data was written to the transfer buffer because the final data block ...
Global Const notArmedError = -10606                   'The specified resource is not armed.
Global Const armedError = -10607                      'The specified resource is already armed.
Global Const noTransferInProgError = -10608           'No transfer is in progress for the specified resource.
Global Const transferInProgError = -10609             'A transfer is already in progress for the specified resource, or the operation ...
Global Const transferPauseError = -10610              'A single output channel in a group may not be paused if the output data ...
Global Const badDirOnSomeLinesError = -10611          'Some of the lines in the specified channel are not configured for the ...
Global Const badLineDirError = -10612                 'The specified line does not support the specified transfer direction.
Global Const badChanDirError = -10613                 'The specified channel does not support the specified transfer direction, ...
Global Const badGroupDirError = -10614                'The specified group does not support the specified transfer direction.
Global Const masterClkError = -10615                  'The clock configuration for the clock master is invalid.
Global Const slaveClkError = -10616                   'The clock configuration for the clock slave is invalid.
Global Const noClkSrcError = -10617                   'No source signal has been assigned to the clock resource.
Global Const badClkSrcError = -10618                  'The specified source signal cannot be assigned to the clock resource.
Global Const multClkSrcError = -10619                 'A source signal has already been assigned to the clock resource.
Global Const noTrigError = -10620                     'No trigger signal has been assigned to the trigger resource.
Global Const badTrigError = -10621                    'No trigger signal has been assigned to the trigger resource.
Global Const preTrigError = -10622                    'The pretrigger mode is not supported or is not available in the current ...
Global Const postTrigError = -10623                   'No posttrigger source has been assigned.
```

```
Global Const delayTrigError = -10624                 'The delayed trigger mode is not supported or is not available in the current ...
Global Const masterTrigError = -10625                'The trigger configuration for the trigger master is invalid.
Global Const slaveTrigError = -10626                 'The trigger configuration for the trigger slave is invalid.
Global Const noTrigDrvError = -10627                 'No signal has been assigned to the trigger resource.
Global Const multTrigDrvError = -10628               'A signal has already been assigned to the trigger resource.
Global Const invalidOpModeError = -10629             'The specified operating mode is invalid, or the resources have not been ...
Global Const invalidReadError = -10630               'The parameters specified to read data were invalid in the context of the ...
Global Const noInfiniteModeError = -10631            'Continuous input or output transfers are not allowed in the current operating ...
Global Const someInputsIgnoredError = -10632         'Certain inputs were ignored because they are not relevant in the current ...
Global Const invalidRegenModeError = -10633          'The specified analog output regeneration mode is not allowed for this ...
Global Const noContTransferInProgressError = -10634  'No continuous (double buffered) transfer is in progress for the specified ...
Global Const invalidSCXIOpModeError = -10635         'Either the SCXI operating mode specified in a configuration call is invalid, ...
Global Const noContWithSynchError = -10636           'You cannot start a continuous (double-buffered) operation with a synchronous ...
Global Const bufferAlreadyConfigError = -10637       'Attempted to configure a buffer after the buffer had already been configured. ...
Global Const badClkDestError = -10638                'The clock cannot be assigned to the specified destination.
Global Const rangeBadForMeasModeError = -10670       'The input range is invalid for the configured measurement mode.
Global Const autozeroModeConflictError = -10671      'Autozero cannot be enabled for the configured measurement mode.
Global Const badChanGainError = -10680               'All channels of this board must have the same gain.
Global Const badChanRangeError = -10681              'All channels of this board must have the same range.
Global Const badChanPolarityError = -10682           'All channels of this board must be the same polarity.
Global Const badChanCouplingError = -10683           'All channels of this board must have the same coupling.
Global Const badChanInputModeError = -10684          'All channels of this board must have the same input mode.
Global Const clkExceedsBrdsMaxConvRateError = -10685 'The clock rate exceeds the board's recommended maximum rate.
Global Const scanListInvalidError = -10686           'A configuration change has invalidated the scan list.
Global Const bufferInvalidError = -10687             'A configuration change has invalidated the acquisition buffer, or an acquisition ...
Global Const noTrigEnabledError = -10688             'The number of total scans and pretrigger scans implies that a triggered ...
Global Const digitalTrigBError = -10689              'Digital trigger B is illegal for the number of total scans and pretrigger ...
Global Const digitalTrigAandBError = -10690          'This board does not allow digital triggers A and B to be enabled at the ...
Global Const extConvRestrictionError = -10691        'This board does not allow an external sample clock with an external scan ...
Global Const chanClockDisabledError = -10692         'This board does not allow an external sample clock with an external scan ...
Global Const extScanClockError = -10693              'You cannot use an external scan clock when doing a single scan of a single ...
Global Const unsafeSamplingFreqError = -10694        'The scan rate is above the maximum or below the minimum for the hardware, ...
Global Const DMAnotAllowedError = -10695             'You have set up an operation that requires the use of interrupts.  DMA ...
Global Const multiRateModeError = -10696             'Multi-rate scanning cannot be used with the AMUX-64, SCXI, or pretriggered ...
Global Const rateNotSupportedError = -10697          'Unable to convert your timebase/interval pair to match the actual hardware ...
Global Const timebaseConflictError = -10698          'You cannot use this combination of scan and sample clock timebases for ...
Global Const polarityConflictError = -10699          'You cannot use this combination of scan and sample clock source polarities ...
Global Const signalConflictError = -10700            'You cannot use this combination of scan and convert clock signal sources ...
```

```
Global Const noLaterUpdateError = -10701          'The call had no effect because the specified channel had not been set ...
Global Const prePostTriggerError = -10702         'Pretriggering and posttriggering cannot be used simultaneously on the ...
Global Const noHandshakeModeError = -10710        'The specified port has not been configured for handshaking.
Global Const noEventCtrError = -10720             'The specified counter is not configured for event-counting operation.
Global Const SCXITrackHoldError = -10740          'A signal has already been assigned to the SCXI track-and-hold trigger ...
Global Const sc2040InputModeError = -10780        'When you have an SC2040 attached to your device, all analog input channels ...
Global Const outputTypeMustBeVoltageError = -10781   'When you have an SC2040 attached to your device, all analog input channels ...
Global Const sc2040HoldModeError = -10782         'The specified operation cannot be performed with the SC-2040 configured ...
Global Const calConstPolarityConflictError = -10783  'Calibration constants in the load area have a different polarity from ...
Global Const timeOutError = -10800                'The operation could not complete within the time limit.
Global Const calibrationError = -10801            'An error occurred during the calibration process.  Possible reasons for ...
Global Const dataNotAvailError = -10802           'The requested amount of data has not yet been acquired.
Global Const transferStoppedError = -10803        'The on-going transfer has been stopped.  This is to prevent regeneration ...
Global Const earlyStopError = -10804              'The transfer stopped prior to reaching the end of the transfer buffer.
Global Const overRunError = -10805                'The clock rate is faster than the hardware can support.  An attempt to ...
Global Const noTrigFoundError = -10806            'No trigger value was found in the input transfer buffer.
Global Const earlyTrigError = -10807              'The trigger occurred before sufficient pretrigger data was acquired.
Global Const LPTcommunicationError = -10808       'The trigger occurred before sufficient pretrigger data was acquired.
Global Const gateSignalError = -10809             'Attempted to start a pulse width measurement with the pulse in the phase ...
Global Const internalDriverError = -10810         'An unexpected error occurred inside the driver when performing this given ...
Global Const softwareError = -10840               'The contents or the location of the driver file was changed between accesses ...
Global Const firmwareError = -10841               'The firmware does not support the specified operation, or the firmware ...
Global Const hardwareError = -10842               'The hardware is not responding to the specified operation, or the response ...
Global Const underFlowError = -10843              'Because of system and/or bus-bandwidth limitations, the driver could not ...
Global Const underWriteError = -10844             'Your application was unable to deliver data to the background generation ...
Global Const overFlowError = -10845               'Because of system and/or bus-bandwidth limitations, the driver could not ...
Global Const overWriteError = -10846              'Your application was unable to retrieve data from the background acquisition ...
Global Const dmaChainingError = -10847            'New buffer information was not available at the time of the DMA chaining ...
Global Const noDMACountAvailError = -10848        'The driver could not obtain a valid reading from the transfer-count register ...
Global Const OpenFileError = -10849               'The configuration file or DSP kernel file could not be opened.
Global Const closeFileError = -10850              'Unable to close a file.
Global Const fileSeekError = -10851               'Unable to seek within a file.
Global Const readFileError = -10852               'Unable to read from a file.
Global Const writeFileError = -10853              'Unable to write to a file.
Global Const miscFileError = -10854               'An error occurred accessing a file.
Global Const osUnsupportedError = -10855          'NI-DAQ does not support the current operation on this particular version ...
Global Const osError = -10856                     'An unexpected error occurred from the operating system while performing ...
```

```
Global Const internalKernelError = -10857          ' An unexpected error occurred inside the kernel of the device while performing ...
Global Const hardwareConfigChangedError = -10858    ' The system has reconfigured the device and has invalidated the existing ...
Global Const updateRateChangeError = -10880        ' A change to the update rate is not possible at this time because 1) when ...
Global Const partialTransferCompleteError = -10881  ' You cannot do another transfer after a successful partial transfer.
Global Const daqPollDataLossError = -10882         ' The data collected on the Remote SCXI unit was overwritten before it could ...
Global Const wfmPollDataLossError = -10883         ' New data could not be transferred to the waveform buffer of the Remote ...
Global Const pretrigReorderError = -10884          ' Could not rearrange data after a pretrigger acquisition completed.
Global Const overLoadError = -10885                ' The input signal exceeded the input range of the ADC.
Global Const gpctrDataLossError = -10920           ' One or more data points may have been lost during buffered GPCTR operations ...
Global Const chassisResponseTimeoutError = -10940   ' No response was received from the Remote SCXI unit within the specified ...
Global Const reprogrammingFailedError = -10941      ' Reprogramming the Remote SCXI unit was unsuccessful. Please try again.
Global Const invalidResetSignatureError = -10942    ' Reprogramming the Remote SCXI unit was unsuccessful. Please try again.
Global Const chassisLockupError = -10943           ' The interrupt service routine on the remote SCXI unit is taking longer ...

'***********************************************************************'
'*
'* Mapping of old errors and warnings to new
'*
'* Warnings
'*
'* dupIOaddrRange            -(dupAddressError)
'* dupIntLevels              -(dupIntError)
'* dupDMALevels              -(dupDMAError)
'* readOutputPort            -(badChanDirError)
'* calibrationErr            -(calibrationError)
'* noPreTrigUnwrap           -(memFullError)
'* relatedPortBusy           -(relatedLCGBusyError)
'* readOutputLine            -(badDirOnSomeLinesError)
'* outOnSomeInLines          -(badDirOnSomeLinesError)
'* inOnSomeOutLines          -(badDirOnSomeLinesError)
'* simulOpAcrossChips        -(invalidOpModeError)
'* overWriteBeforeCopy       -(overWriteError)
'* pageBreakinWFbuf          -(memPageError)
'* wrongNumConfigBytes       -(noSetupError)
'* DMAReprogramming          -(memPageError)
'* SCXImoduleTypeConflict    -(SCXIModuleTypeConflictError)
'* notEnoughExtMem           -(memFullError)
'* inputModeConflict         -(invalidOpModeError)
'* SCXIConfigWarning         -(SCXIConfigError)
'* messageIntervalTooLong    -(badDAQEventError)
'* logicalDeviceWarning      -(badDeviceError)
'* calConstPolarityConflict  -(calConstPolarityConflictError)
'* irqConflict               -(dupIntError)
'* dmaConflict               -(dupDMAError)
'* jumperlessBoardWarning    -(jumperlessBoardError)
'* gpctrDataLossWarning      -(gpctrDataLossError)
'*
'*
'* Errors
'*
'* notOurBrdErr              unknownDeviceError
'* badBrdNumErr              badDeviceError
'* badGainErr                badGainError
'* badChanErr                badChanError
```

* noSupportErr	deviceSupportError
* badPortErr	badChanError
* badOutPortErr	badChanDirError
* noLatchModeErr	noHandshakeModeError
* noGroupAssign	noGroupAssignError
* badInputValErr	invalidValueError
* timeOutErr	timeOutError
* outOfRangeErr	badRangeError
* daqInProgErr	transferInProgError
* counterInUseErr	counterBusyError
* noDAQErr	noTransferInProgError
* overFlowErr	overFlowError
* overRunErr	overRunError
* badCntErr	badCountError
* brdTypeErr	deviceSupportError
* noCountOpErr	noEventCtrError
* ctrReservedErr	sysOwnedRsrcError
* portAssignToGrp	groupAssignError
* noPortAssignErr	noGroupAssignError
* badGrpDirErr	badGroupDirError
* noGrpBlockInProg	noTransferInProgError
* grpBlockInProg	transferInProgError
* setLatchWGrpCall	invalidValueError
* laterIntUpdateNotSet	noLaterUpdateError
* wfInProgErr	transferInProgError
* noWfLoadErr	noWriteError
* noWfInProgErr	noTransferInProgError
* badPreTrigCntErr	badPretrigCountError
* buffNotFullErr	earlyTrigError
* prePostTrigErr	prePostTriggerError
* extConvErr	extConvRestrictionError
* badSigDirErr	badLineDirError
* noDbDaqErr	noContTransferInProgError
* overWriteErr	overWriteError
* memErr	memFullError
* noConfigFile	configFileError
* badGrpSize	badGroupError
* intLevelInUse	intInUseError
* DMAChanInUse	DMAInUseError
* multSourceInputErr	multConnectError
* lowScanIntervalErr	lowScanIntervalError
* noConnectionErr	noConnectError
* noPGInProg	noTransferInProgError
* PGInProg	transferInProgError
* grpRateErr	counterBusyError
* extGateErr	invalidOpModeError
* openFileErr	openFileError
* writeFileErr	writeFileError
* noDbWvfmErr	noTransferInProgError
* oldDataErr	transferStoppedError
* dataNotAvailErr	dataNotAvailError
* DMATransferCntNotAvail	noDMACountAvailError
* noLabScanErr	noTransferInProgError
* dbOpErr	noContWithSynchError
* DMADisabledErr	noDMAAvailError
* invalidConfigErr	cmosConfigError
* brdIsArmedErr	armedError
* clockSourceErr	multClkSrcError
* noSetupErr	noSetupError
* extConvDrvErr	multClkSrcError
* triggerSourceErr	badTrigError
* noArmErr	notArmedError
* intDisabledErr	noIntAvailError
* keyNotFoundErr	configFileError
* noTrigEnabledErr	preTrigError
* digPortReserved	sysOwnedRsrcError

* RTSIlineInUseErr	sysOwnedRsrcError
* dacUpdateRTSIinUseErr	sysOwnedRsrcError
* noRTSIlineAvailErr	noLineAvailError
* preTrigScansErr	badPretrigCountError
* postTrigScansErr	badPosttrigCountError
* scanRateErr	badIntervalError
* invalidGetErr	invalidReadError
* calInputOutOfRange	badExtRefError
* EEPROMaddrErr	EEPROMreadError
* EEPROMresponseErr	EEPROMreadError
* EEPROMreadErr	EEPROMreadError
* EEPROMwriteErr	EEPROMwriteError
* calResponseErr	calibrationError
* calConvergeErr	calibrationError
* calDACerr	calibrationError
* externalCalRefErr	badExtRefError
* internalCalRefErr	hardwareError
* badOutLineErr	badLineDirError
* relatedPortAssignToGrpBusy	relatedLCGBusyError
* dacUpdateErr	underFlowError
* muxMemFullErr	muxMemFullError
* interlvdDataAlignErr	memAlignmentError
* cannotAlignBufErr	memAlignmentError
* cannotLockBufErr	memLockError
* cannotPageLockErr	memPageLockError
* invalidChassisIDErr	badChassisIDError
* invalidModuleSlotErr	badModuleSlotError
* configFileErr	configFileError
* outdatedVDMADErr	oldDriverError
* ctrRTSINotAvailErr	lineBusyError
* dacUpdateRTSINotAvailErr	lineBusyError
* SCXIConfigErr	SCXIConfigError
* noDbDigErr	noTransferInProgError
* DbDigPartialComplete	transferStoppedError
* SCXITrackHoldErr	SCXITrackHoldError
* wvfmGrpAssignErr	groupAssignError
* chanNotAssignedGrpErr	noGroupAssignError
* grpLoadErr	groupWriteError
* loadAfterStartErr	activeWriteError
* noUpdateRateErr	noClkSrcError
* chanPauseErr	transferPauseError
* DSPInitFailure	DSPInitError
* DSPDataPathInUse	DSPDataPathBusyError
* DSPDAQErr	internalKernelError
* DSPreserved3	badErrorCodeError
* DSPreserved4	badErrorCodeError
* DSPreserved5	badErrorCodeError
* SCXICommErr	communicationsError
* invalidOpModeErr	invalidSCXIOpModeError
* moduleNotSupported	SCXIModuleNotSupportedError
* DAQboardNotSupported	deviceSupportError
* noNIDAQLibErr	noDriverError
* noNIDAQFuncErr	functionNotFoundError
* incompatibleVISRDErr	oldDriverError
* port1InLatchedModeErr	relatedLCGBusyError
* invalidMemRegionErr	memLockError
* fifoModeErr	fifoModeError
* cannotFreeMemErr	memConfigError
* memNotLockedErr	memConfigError
* invalidWinHandleErr	invalidWinHandleError
* trigEventNotAvailErr	DMANotAllowedError
* memTypeNotSupportedErr	memConfigError
* badChanStrErr	syntaxError
* parseErr	syntaxError
* noSuchMessageErr	noSuchMessageError
* badChanTypeErr	badChanError

```
* badTrigValErr              badDAQEventError
* notOurDSPHandleErr         invalidDSPHandleError
* NIDAQInternalErr           internalDriverError
* preTrigReorderErr          pretrigReorderError
* badCtrErr                  badCounterError
* invalidCtrErr              badCounterError
* timedMsgInUseErr           counterBusyError
* invDAQModeTimedMsgErr      DMANotAllowedError
* lptCommunicationErr        LPTcommunicationError
* multiRateAMUXErr           multiRateModeError
* multiRatePreTrigErr        multiRateModeError
* functionNotLinkedErr       internalDriverError
* scanIntervalTooLongErr     badIntervalError
* sampleIntervalTooLongErr   badIntervalError
* updateIntervalTooLongErr   badIntervalError
* gpctrBadApplicationErr     gpctrBadApplicationError
* gpctrBadCounterNumberErr   gpctrBadCounterNumberError
* gpctrBadParamValueErr      gpctrBadParamValueError
* gpctrBadParamIdErr         gpctrBadParamIdError
* gpctrBadEntityIdErr        gpctrBadEntityIdError
* gpctrBadActionErr          gpctrBadActionError
* gpctrBadGateSignalErr      gateSignalError
* gpctrNotArmedErr           noSetupError
* gpctrNotResetErr           counterBusyError
* gpctrNotProgrammedErr      noSetupError
* gpctrApplicationNotSetErr  noSetupError
* gpctrBufferNotConfiguredErr  bufferInvalidError
* gpctrCantChangeParameterErr  counterBusyError
* lptProtocolNotSupported    LPTcommunicationError
* rateNotSupportedErr        rateNotSupportedError
* timebaseConflictErr        timebaseConflictError
* polarityConflictErr        polarityConflictError
* signalConflictErr          signalConflictError
* baseAddressErr             baseAddressError
* interruptLevel1Err         badErrorCodeError
* interruptLevel2Err         badErrorCodeError
* dmaChannel1Err             dmaChannel1Error
* dmaChannel2Err             dmaChannel2Error
* openSCManagerErr           badErrorCodeError
* openNIDAQServiceErr        badErrorCodeError
* startNIDAQServiceErr       badErrorCodeError
* criticalResourceConflictErr  badErrorCodeError
* jumperlessBoardErr         jumperlessBoardError
* reservedPinErr             reservedPinError
* bufferNotInterleavedErr    bufferNotInterleavedError
* gpctrInUseErr              counterBusyError
* gpctrDataLossErr           gpctrDataLossError
* updateRateChangeErr        updateRateChangeError
* gpctrBufferConfiguredErr   bufferAlreadyConfigError
* gpctrBufOprnNotInProgErr   noTransferInProgError
* badFilterFreqErr           badFilterCutoffError
* sc2040HoldModeErr          sc2040HoldModeError
* sc2040InputModeErr         sc2040InputModeError
* noSC2040ConfigErr          noSetupError
* DAQCardConfigErr           DAQCardConfigError
* partialTransferCompleteErr   partialTransferCompleteError
* DMABufferAlignmentErr      memAlignmentError
* outputTypeMustBeVoltageErr   outputTypeMustBeVoltageError
* osUnsupportedErr           osUnsupportedError
* osErr                      osError
**********************************************************************'
```

NIDEX32 MODULE

```
**********************************************************************
```

```
'* TITLE:     NIDEx32.bas
'*           Header for supporting code module for NI-DAQ Examples
'*           (32-bit Visual Basic version)
'*
'* DESCR:     This header file is to be used with any NI-DAQ example
'*           program.
'*************************************************************************/

'NOTE: must also use nidaq32.bas
'     and nidaqcns.bas

'*
'* Constants
'*

'* for 'lType'
Global Const WFM_DATA_U8 = 0
Global Const WFM_DATA_I16 = 2
Global Const WFM_DATA_F64 = 4
Global Const WFM_DATA_U32 = 7

'* internal constants - change if needed...
Global Const WFM_PERIODS = 10
Global Const WFM_MIN_PTS_IN_PERIOD = 2
Global Const WFM_U8_MODULO = 256
Global Const WFM_I16_AMPL = 2047
Global Const WFM_F64_AMPL = 4.99


'* error return codes for NIDAQPlotWaveform and NIDAQMakeBuffer
'* these error codes are consistent with CVI error codes
Global Const NIDAQEX_INVALID_BUFFER = -12
Global Const NIDAQEX_INVALID_NUMPTS = -14
Global Const NIDAQEX_INVALID_TYPE = -53


'*
'* Function prototypes
'*

Declare Function NIDAQPlotWaveform Lib "nidex32.dll" (pvBuffer As Any, ByVal lNumPts&, ByVal lType&)
As Integer
Declare Function NIDAQMakeBuffer Lib "nidex32.dll" (pvBuffer As Any, ByVal lNumPts&, ByVal lType&) As
Integer
Declare Function NIDAQErrorHandler Lib "nidex32.dll" (ByVal iStatus%, ByVal strFuncName$, ByVal
iIgnoreWarning%) As Integer
Declare Function NIDAQDelay Lib "nidex32.dll" (ByVal dSec#) As Integer
Declare Function NIDAQYield Lib "nidex32.dll" (ByVal iYieldMode%) As Integer
Declare Function NIDAQMean Lib "nidex32.dll" (pvBuffer As Any, ByVal lNumPts&, ByVal lType&, dMean#)
As Integer
Declare Function NIDAQWaitForKey Lib "nidex32.dll" (ByVal dTimeLimit#) As Integer
```

ABSTRACT

The objectives of the work reported here were to design and integrate a communications interface and software procedures (i.e., algorithms) for image processing for a Helmet Mounted Display (HMD) image tester. This is a continuation of a previous effort entitled “Preliminary Design of an Image Tester for Helmet Mounted Display.”

The proposed image quality tester consists of hardware (including camera, lenses, sensors, and fixtures), and software for image capture and analysis. The interface and image processing algorithms are essential components of this system. The interface bridges the gap between hardware devices and software applications, and thus makes information integration possible. The algorithms process, analyze, and characterize the test pattern information generated by a Helmet Mounted Display (HMD).

An interface was designed to probe sensor information and coordinate/synchronize image capture and analysis events. A set of three limited switches was utilized to indicate the presence of an HMD, the position of an HMD relative to a wide-angle camera, and the
Block 19 continued

position of an HMD relative to a narrow-angle camera. These switches are connected to a data acquisition card (DAQCard-DIO-24) using designed circuitry. The sensor on/off states are recorded by the card registers. Software routines (i.e., algorithms) were designed and developed to probe the register status, and then use this information to coordinate/synchronize image characterization events. In order to enhance the flexibility and reduce the complexity of the existing image capture application, a new image capture module was designed.

In designing the algorithms, issues such as data collection steps, design specifications, and noise generation were taken into consideration. Three HMD units were utilized to capture image data. Images with noise—such as displacement and variations in orientation and focus—were captured. Statistical approaches such as correlation coefficients and regression analysis were utilized to probe the relationships between performance/image related variables (such as focus) and image gray level variation. Knowledge of such relationships enables the use of image variables to verify and/or predict control variables such as focus resolution. Image measurement specifications were developed based on analysis of the collected image data. Algorithms for detecting four vertical lines, center point, focus, and boundary are proposed. Examples are given to illustrate how the algorithms work and screenshots of images before and after image processing are shown.

EXAMPLE IMPLEMENTATION B

(The following is a substantial duplicate of Hsieh, et al., "Preliminary Design of an Image Quality Tester For Helmet-Mounted Displays," USAARL Report No. 2000-08 (November 1999), the content of which is hereby incorporated by reference in its entirety.

USAARL Report No. 2000-08

Preliminary Design of an Image Quality Tester for Helmet-Mounted Displays

by

Sheng-Jen Hsieh

Texas A&M University

and

Clarence E. Rash

Aircrew Health and Performance Division

and

Thomas H. Harding
Howard H. Beasley
John S. Martin

UES, Inc.

December 1999

Approved for public release, distribution unlimited.

U.S. Army Aeromedical Research Laboratory
Fort Rucker, Alabama 36362-0577

Notice

Qualified requesters

Qualified requesters may obtain copies from the Defense Technical Information Center (DTIC), Cameron Station, Alexandria, Virginia 22314. Orders will be expedited if placed through the librarian or other person designated to request documents from DTIC.

Change of address

Organizations receiving reports from the U.S. Army Aeromedical Research Laboratory on automatic mailing lists should confirm correct address when corresponding about laboratory reports.

Disposition

Destroy this document when it is no longer needed. Do not return it to the originator.

Disclaimer

The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documents. Citation of trade names in this report does not constitute an official Department of the Army endorsement or approval of the use of such commercial items.

Table of contents

	Page
Introduction	91
Functionality and operating process	91
Methodology	92
Image capture hardware specification	92
Test pattern features investigation	95
Software prototype design	96
Image capture module	97
Image analysis and interpretation module	98
Algorithm design	98
Testing and validation	100
Hardware package design	100
Conclusions and future directions	101
References	102
Appendix A. List of manufacturers	103
Appendix B. Software prototype program	104

List of figures

B1. The IHU of the AH-64 IHADSS	91
B2. The IHADSS HDU	91
B3. Display size	92
B4. Test pattern from the IHADSS HMD	92

Table of contents (continued)
List of figures (continued)

	Page
B5. Flow chart for the HMD prototype tester operation	92
B6. Experimental setup for camera sensitivity analysis	93
B7. Sampling locations on the test pattern	93
B8. Plot of photometer and CCD camera data	94
B9. Setup for test pattern measurement	95
B10. Test pattern design based on measurement results	95
B11. Replicated test pattern image	95
B12. Measurement of luminance of the center lines	96
B13. Center lines measurement with varied focus	96
B14. Designed test pattern with focus on the center lines	96
B15. Opening screen of prototype software	97
B16. Image capture module	97
B17. Image capture component	97
B18. Image processing component	97
B19. Image analysis and interpretation module	97
B20. Tilted test pattern binary images from image analysis module	100
B21. Overall testing results of an HMD	100
B22. Tilted test pattern before (left) and after (right) Sober edge detection	100
B23. Investigation of CCD image capture arrangement	101
B24. CAD concept of prototype hardware design	101

List of tables

Measured data and correlation coefficient from photometer and CCD camera 94

Introduction

Helmet-mounted displays (HMDs) are a gateway to the pilot for viewing pilotage and fire control imagery. In Army aviation, the AH-64 Apache helicopter uses an HMD system known as the Integrated Helmet and Display Sighting System (IHADSS). The IHADSS consists of various electronic components and a helmet/display system called the Integrated Helmet Unit (IHU). The IHU (Figure B1) includes a helmet, visor housings with visors, miniature cathode ray tube (CRT), and helmet display unit (HDU). The HDU serves as an optical relay device which conveys the image formed on the CRT through a series of lenses, off a beamsplitter (called a combiner), and into the aviator's right eye (Figure B2). The CRT is 1 inch in diameter and uses a P-43 phosphor. The combiner is a multilayer dichroic filter which is maximized for reflectance at the peak emission of the P-43 phosphor.

The U.S. Army is currently developing the next generation reconnaissance aircraft, the RAH-66 Comanche. This aircraft will incorporate an HMD which will be binocular in design. While its final design is still in review, it will basically consist of two image sources (either miniature CRTs or liquid crystal displays) with two sets of optics, delivering imagery to both eyes.

See
Figure B1. The IHU of the AH-64 IHADSS.

Currently, there is no existing image quality tester for HMD validation in the field. To maintain system integrity and readiness, and to provide pilots with validated pilotage, navigation, and fire control imagery, there is a need to design and construct an image quality testing tool for the HMD. The objective of this study is to propose and test a design concept for an image quality tester for HMD subsystems. The tester can be used as a validation tool to verify settings for regular flight missions and for preventive maintenance tasks. The first prototype tester will be designed for the AH-64's IHADSS HMD.

See
Figure B2. The IHADSS HDU.

Functionality and operating process

The proposed tester will allow pilots and maintenance personnel to validate the image quality of an HMD. Basic required characteristics include (1) simple design, (2) ease of use, (3) robustness, and (4) accuracy for operations and maintenance. The prototype should be small enough to fit into a brief case, which would include a lap-top, image capture system, and power supply pack.

The IHADSS HMD has a monocular 30-degree vertical by 40-degree horizontal field-of-view (FOV). Future HMDs most likely will have larger FOVs and be binocular in design. HMD corner obscurations are generally permissible and symmetrical for the IHADSS, as illustrated in Figure B3. Since hardware changes to

the various aircraft electronics will not be allowed, image quality validation must be performed using manufacturer built-in test patterns. The built-in test pattern of the IHADSS HMD is used as the inspection specification on which the first tester will be based. The test pattern shows strips of gray opposed along the vertical center lines. Each strip contains 8 to 10 shades of gray, depending on the contrast ratio. Adjacent shades have a square root of 2 differential of brightness. Figure B4 is a snapshot of the test pattern captured from the IHADSS HMD. For more detailed discussion of the HMD test pattern features, see the Honeywell, Inc. study guide (1985) and Harding et al. (1995). For testing this test pattern, the inspection features used by the image quality tester prototype will include (1) four vertical center lines, (2) one horizontal center line, (3) two gray shade patterns (with 8 to 10 shades), and (4) a boundary box.

See
Figure B3. Display size.

See
Figure B4. Test pattern the IHADSS HMD.

Based on the design objectives and inspection procedures, the tester operation procedures are as follows: (1) the pilot adjusts the HMD settings and passes the HDU to the crew chief; (2) the crew chief inserts the HMD into a fixture; (3) the system examines the center and horizontal line features of the test pattern using a narrow-angle lens; (4) the system inspects the test pattern for image displacement and/or disorientation; (5) the system examines the number of gray-shades, the focus, luminance, and boundary lines, using a 42-degree wide-angle lens; and (6) the system generates a final report. Figure B5 shows a flow chart for the proposed operation procedures.

See
Figure B5. Flow chart for HMD prototype tester operation.

Methodology

This study involved designing and testing (1) the hardware specification for image capture, (2) the test pattern inspection features, (3) the software prototype, and finally (4) the hardware prototype. Experiments and statistical analysis tools were applied throughout the design process.

Image capture hardware specifications

To determine the needed camera and lens specification for test pattern image capture, experiments were conducted to verify the sensitivity of a candidate camera. The camera and a Photo Research (Appendix A) model 1980 photometer were mounted using a reconfigurable optical fixture and bench accessories and were used to capture an electronically generated gray shade test pattern. Figure B6 illustrates the experimental setup. The luminance of the test pattern image was registered by the charged couple device (CCD) camera (and image capture card) and the photometer.

Figure B7 shows the locations where data were sampled from the test pattern. These data were measured from a fixed position along a horizontal line across the entire test pattern. Three measurements were taken from each region. An observation resulting from the experiment was that the luminances of the gray shades presented in the test pattern were not linearly distributed between 0 and 255. The differential of luminance for adjacent shades was greater than an approximate square root of 2. A statistical analysis was performed on these data. Results indicated that the luminance levels measured by the photometer were consistent with data from the camera and image capture card up to and including the 7th gray shade. It can be seen that the CCD saturated after the 7th gray shade area. To prevent this, the aperture of the CCD would have to be adjusted. If only the first seven gray shades are used in the analysis, correlation is 0.98. The table and Figure B8 record the data collected from both instruments and the statistical analysis results.

See
Figure B6. Experimental setup for camera sensitivity analysis.

See
Figure B7. Sampling locations on the test pattern.

Table.
Measured data and correlation coefficient from photometer and CCD camera.

Gray shade	Photometer luminance readings			CCD gray level readings		
1	3.25	3.32	3.35	7	7	7
2	7.47	7.51	7.46	25	25	25
3	17.07	16.99	16.99	65	65	65
4	30.51	30.54	30.43	99	99	99
5	48.28	48.24	48.12	146	146	146
6	71.9	71.86	71.81	194	194	194
7	98.35	98.54	98.67	227	227	227
8	127.1	127.2	127.3	230	230	230
9	157.9	158.1	158.0	235	235	235
10	187.4	187.4	187.1	240	240	240
11	221.2	221.4	221.2	242	242	242
12	200.7	200.6	200.6	237	237	237
Luminance vs gray level (7 shades): Correlation = 0.983886; Fisher's z = 2.406549; Probability = 00006						

See
Figure B8. Plot of photometer and CCD camera data.

In an attempt to capture the test pattern image on the IHADSS fully, several different cameras (with standard lenses) were evaluated. However, although the full test pattern could be captured, the details of the four vertical center lines could not be differentiated. Therefore, a decision was made to use a narrow angle lens to zoom in on the center area of the test pattern in order to capture the details of the center lines. HMDs are also used at night; therefore, the prototype tester--specifically the camera--should provide good sensitivity at low luminance levels. First order specifications for the required camera were summarized as follows:

1. Sensitivity: #0.005 lux
2. Focus: To infinity
3. Resolution: > 768 x 498 pixels
4. Focal length: ~½ inch
5. Iris: Manual
6. Fields of view: >40 (H) x 30 (V) degrees and ~5 x 3 degrees

Test pattern features investigation

An additional experiment was conducted to investigate various aspects of capturing the test pattern. Multiple cameras were used since a single camera that met all the desired specifications was not available at the time of this study. Aspects of interest included the size of the pattern, number of different features, relative luminance ratios among features, spatial content of each feature, and number of gray shades. The IHADSS HMD was mounted on the top of the optical post, and the post was fixed on top of a round optical table controlled by a programmable position table. The table was driven by a stepping motor with an accuracy of 1 micron (μm). The test pattern image was projected onto a video monitor for observation. Figure B9 shows the experimental setup. The entire test pattern image from the HMD was captured and constructed through a series of mini steps in the horizontal and vertical directions. The overall picture was approximately 38 x 29 degrees, which was close to the specification in the study guide (Honeywell, Inc.,1985). The center line occupied approximately 0.5 degree out of 38 degrees. There were two strips with 10 to 12 gray shades mirrored opposite the center lines. Figure B10 shows the structure of the IHADSS test pattern. A series of images were taken to probe the content of each gray shade in terms of luminance. Based on the observed information, a series of image files was constructed and used as an image profile for purposes of the software prototype development. Figure B11 displays this replicated test pattern image.

See
Figure B9. Setup for test pattern measurement.

See
Figure B10. Test pattern design based on measurement results.

See
Figure B11. Replicated test pattern image.

A similar experiment was conducted to detail the center lines within the test pattern. Figure B12 shows the luminance scan measurements for the center lines. The four peaks represent the four center lines which are spread out over 0.8 degree from valley to valley and 0.4 degree peak to peak. The average peak width is about 0.0969 degree and the average distance between peaks is about 0.1347 degree. Note: A measurement of 1 degree is about 485 μm in the object plane.

Another experiment was conducted to probe the state of the center lines when the HMD is in focus and not in focus. Varied focus values of -1 to 1 diopter of CRT were applied. Measurements of the four vertical center lines were taken. An interesting finding was, when the HMD was in focus, the ratio of luminances between bottom to mid-peak (B) and peak to valley (A) was close to 1. However, when the setting was not in focus, the B:A ratio was less than one. Figure B13 documents these

observations and illustrates the concept. Findings from the above experiments, such as measurements, luminance ratios, and the content of each feature within the test pattern, were used to create a test pattern image using graphics software. Figure B14 shows an image of such a test pattern using a 5 X 4 degree lens to focus on the center lines of the test pattern. In addition, the ratio of the square root of 2 luminance difference was used to design gray shades ranging from 0 to 255 gray levels.

See
Figure B12. Measurement of luminance of the center lines.

See
Figure B13. Center lines measurement with varied focus.

To emulate potential human errors in setting up the HMD, a set of parameters (including brightness, orientation, spatial adjustment, and contrast) were manipulated and the resulting images captured. These images were used as a basis for creating new image files. These designed images were used to test the software prototype. The experiments were carried out using similar methods. For example, to measure the potential displacement of the test pattern, a camera was mounted facing the HMD. The test pattern was projected onto a video monitor by means of a personal computer (PC). Measurements were taken before and after the spatial adjustments. The maximum adjustments in the upward, downward, left and right directions were 3.57, 2.98, 4.90 and 4.90 degrees, respectively, based on an FOV of 40 x 31 degrees (Harding et al., 1995).

See
Figure B14. Designed test pattern with focus on the center lines.

Software prototype design

The software prototype was designed to capture, analyze, and interpret the image against test pattern features such as the four center lines and number of gray shades. Accordingly, the prototype design will require three modules--image acquisition, image analysis and interpretation--as well as on-line user help. Figure B15 shows the modules involved in the prototype. Visual Basic (VB) was used to develop the prototype because of its flexibility in linking and embedding with other commercial software and because it was a powerful toolbox for rapidly prototyping a complicated window. In the following sections, we describe the functionality of each module and how the modules are integrated. Algorithms developed to interpret the image follow. Finally, testing and validation of the code is addressed. The source code for the program can be found in Appendix B.

Image capture module

The VB Object Linking and Embedding (OLE) capability allows integration of other programs. In this case, the image capture graphics program served as an object which was linked into the VB main program. The graphics program was launched by activating the linked object. Once the object had been activated, the VB main program allowed the user to modify, save, or open documents created by the graphic program in VB's integrated design environment (IDE). After the user was done with the image capture graphics program, control was released to the VB environment. The graphics program itself contained three components: the driver used to activate the image capture card and digitize the video signal into a graphics image format (e.g., bitmap or jpeg); an image processing shell which allowed image manipulation (e.g., sharpening and lightening); and an on-line user manual. Figure B16 shows the opening screen for the image capture module. Figures B17 and B18 show image capture and processing subcomponents.

See
Figure B15. Opending screen of prototype software.

See
Figure B16. Image capture module.

See
Figure B17. Image capture. component.

See
Figure B18. Imagine processing component.

Image analysis and interpretation module

The image analysis and interpretation module (1) detects the presence of key features such as center lines within the test pattern, (2) compares selected features against the feature specification, and (3) generates findings. VB components were created to provide these functions and to interface with other modules. A subwindow titled "evaluation criteria" was created to analyze and interpret the captured image from an HMD. A few created algorithms were coded in VB to perform the analysis. Other subwindows, such as a directory box and file list boxes were created to allow retrieval of image files for analysis. Finally, an additional subwindow was designed to display the image currently being analyzed. This module also allows access to other modules via a button control. Figure B19 shows the image analysis and interpretation module.

See
Figure B19. Image analysis and interpretation module.

Algorithm design

Algorithms were developed to detect various features within the test pattern as described earlier. These are described below:

A. Identify the number of center lines.

Step 1. Apply binary image technique to the entire image.
Step 2. Draw multiple lines across X and/or Y axes.
Step 3. Identify mask with feature of B/W. . . W/B.
Step 4. Store the intersection points in an array with multiple dimensions.
Step 5. Construct regression lines based on the points within each dimension.
Step 6. Develop regression lines to compare the parallel property.
Step 7. Average the intersection points around the array to obtain the number of estimated lines.

Note 1: B = black pixel and W = white pixel.
Note 2: Use of linear regression analysis would make the linear mode robust and insensitive to noise presence.
B. Identify the center point.
Step 1. Construct a regression line based on all the intercepted points.
Step 2. Identify the midpoint of an array as a starting point with the feature of W/B/W.
Step 3. Examine neighboring pixels to see if a W/W/W mask exists.
Step 4. If a W/W/W mask exists, stop the procedure; else next step.
Step 5. Check the distance of neighboring pixels from the regression line using a 3 x 3 area.
Step 6. Select the point with the smallest distance from the regression line as the next point.
Step 7. Go to step 3.

C. Identify test pattern orientation and displacement.

Step 1. Compute a theoretical center as point A.
Step 2. Identify the actual center point (based on part B) as point B.
Step 3. Compute the distance between point A and B as d.
Step 4. If d is equal to 0; then the displacement is zero.
Step 5. Construct lines between a given point with points A and B.
Step 6. Compute the angle between lines as orientation angle

D. Identify the number of gray shades within a test pattern.

Step 1. Use the center point as a starting point.
Step 2. Pick five points across the center line that are within the boundary of gray shades.
Step 3. Compute the average gray level of the five points.
Step 4. Store it in one dimension of the array.
Step 5. If the boundary is not reached, move up or down a given distance, and go to Step 3; else

next.

Step 6. Use of square root of two differences to determine the number of gray shades.

E. Identify boundary lines.

Step 1. Use the center point and boundary ratio to determine the region of the image boundary.

Step 2. Locate a starting point white pixel to use for back tracking the rest of the white pixels for each line segment.

F. Identify the focus setting.

Step 1. Use line scan technique to record the pixels along the center lines.

Step 2. Use the B/W/B mask to identify the separation of lines.

Step 3. Compute the ratio of bottom to mid-peak and peak to valley for all four lines.

Step 4. If the ratio is approximately one, we may conclude that the focus setting is good; or else check the focus setting

Other methods for center point detection exist. However, these were deemed less appropriate for this application. For instance:

Alternate approach #1:

```
                          b b b b
Step 1.  Use of the mask of  bwwwwwwb
                          b b b b
```

Note: If the orientation of the image is unknown, this method can be time consuming.

Alternate approach #2:

Step 1. Find the center point of each line.

Step 2. Use the averaging method to find the center of all the centers.

Note: This method involves more steps than the proposed one, because you must first find the center of each line and there are four lines to be examined.

Alternate approach #3:

Step 1. Identify the boundary of the image.

Step 2. Use the center of gravity method to find the center of the image.

Figure B20(a-d) shows screens from the image analysis module. Figure B20a shows a binary image of the test pattern after the binary image technique had been applied to the test pattern captured from the HMD. Figure B20b shows the four center lines that were identified from the binary image (Figure B20a). After the center lines had been identified, the image analysis module identified the center point of the image. Figure B20c shows the coordinates (y only shown) of the center point. The image analysis module then determined if the image was tilted or not. Figure B20d displays the tilt angle of the image. The analysis results are summarized and displayed in Figure B21. A primary feature of the image analysis module is to

identify features present in the captured test pattern. The "Sober operator," a well known edge detection technique, is used to identify the boundaries of the features and, thereby, allow the analysis module to determine whether or not the required features are present in the captured test pattern image. Figure B22 shows the same image before and after the Sober operator is applied.

Testing and validation

To verify the accuracy of the program, language debugging tools, and split-half and back tracking strategies were imposed throughout the coding process. The program results were compared with the simulation results. For example, to check the accuracy of the constructed regression line, the same data points also were analyzed and compared with the results obtained from a statistics package and hand calculation.

See

Figure B20. Tilted test pattern binary images from image analysis module.

See

Figure B21. Overall testing results of an HMD.

See

Figure B22. Tiled test pattern before (left) and after (right) Sober edge detection.

Hardware package design

A preliminary concept for the hardware package design consists of a display/output module, power supply module, and image capture module. The display/output module should be designed to display/generate inspection results of an HMD test pattern. The power supply module should be designed to provide the voltages needed for the cameras and computer. The design also should include a rechargeable battery pack which will allow the unit to operate in areas without an external power supply. The power supply would be required to provide 12- and 9-volt outputs for the cameras and computer, respectively. Finally, the image capture module must be designed to hold an HMD and two cameras in fixed and contained positions, thereby preventing potential noise that may affect the inspection accuracy. A proposed design is as follows: Two cameras arranged vertically and facing the HMD. [Figure B23 shows one method investigated for aligning the CCD image capture cameras and the HMD.] An inverted HMD fixture will be the most likely one be used in the final concept. The fixture would be mounted with spring return locks on the sides and bottom. The spring return locks will lock the HMD in a fixed position. These locks would prevent the inspection process from continuing if the HMD is not positioned correctly. Once the HMD is in the correct position, a proximity sensor will be used to trigger the image system to start the image capture and interpretation processes. The cover of the image capture module is in the shape of an inverted HMD. It is designed to cover the HMD tightly once it is in the correct

position, and to eliminate any optical noise from the surrounding environment. To enhance the speed of image analysis, an Electronic Programmable Read Only Memory (EPROM) chip, custom programmed to load the executable program for image analysis, could be used. Figure B24 illustrates a preliminary computer aided design (CAD) concept of the hardware prototype design.

Conclusions and future directions

In this project, a design framework for an image quality tester was proposed and evaluated. Functionality and requirements of the tester were identified. Experiments were conducted to test camera sensitivity and to probe aspects of an HMD test pattern using programmable micro-positioning systems and a CCD camera. Test pattern specifications were developed based on these observations. A strategy for image analysis and interpretation was formed, and algorithms were designed to verify the test pattern of a given HMD against the specifications. A prototype software package was written to inspect the test pattern and verify the effectiveness of the algorithms. Finally, a design framework for a concept hardware package was proposed.

See
Figure B23. Investigation of CCD image capture arrangement.

See
Figure B24. CAD concept of prototype hardware design.

To build a brassboard version of a tester, future work must include: (1) fabrication of the hardware design using inverse casting techniques, (2) integration of software and hardware components for a prototype design, (3) field testing of the prototype, (4) incorporation of learning algorithms to increase inspection accuracy, and (5) expansion of functionality from validation to on-line real time interactive adjusting and self-tuning based on a given environmental scenario. From the maintenance perspective, the work can be expanded to self-diagnosis and preventative maintenance (such as life-time prediction).

References

Avionics Systems Group, Military Avionics Division. 1985. Integrated Helmet and Display Sighting System - Study Guide. St. Louis Park, MN: Honeywell, Inc.

Harding, T.H., Beasley, H.H., Martin, J.S. and Rash, C.E. 1995. Physical Evaluation of the Integrated Helmet and Display Sighting System Helmet Display Unit. Fort Rucker, AL: U.S. Army Aero0medical Research Laboratory. USAARL Report No. 95-32.

Appendix A

List of manufacturers.

Photo Research
3000 North Hollywood Way
Burbank, CA 91505

Appendix B

Software prototype program.

ABSTRACT

Helmet-mounted displays (HMD's) provide essential pilotage and fire control imagery information for pilots. However, image quality testers for HMD field performance validation do not currently exist. This research employed techniques from imaging analysis and interpretation, and computer-aided deign/computer-aided manufacturing (CAD/CAM) to investigate a preliminary design for a portable HMD image quality tester.

For this study, a charge coupled device (CCD) camera and lens were selected. Hardware characteristics such as viewing angles in horizontal and vertical positions, dynamic working range at day and night, pixel resolution, focal length, and aperture ratio were evaluated with regard to HMD functionality. Experiments to evaluate camera sensitivity and test pattern merits were conducted using a programmable micro-positioning system, CCD camera, optical fixtures and benches. Next, the relative ratio among features within the image profile was established and an ideal image profile and evaluation criteria were established based on the experimental results. Third, image processing algorithms and techniques, such as edge detection, were developed and applied in test pattern feature detection. A software prototype including modules for image capture, image analysis and interpretation, and user manuals was developed. Finally, a concept hardware package design is proposed. This design incorporates a notebook computer with flat panel display to interface with the camera and software prototype; and fixtures for the HMD, camera, computer, and power supply. This design will allow the tester to be used in the field.

The foregoing described embodiments depict different components contained within, or connected with, different other components. It is to be understood that such depicted architectures are merely exemplary, and that in fact many other architectures can be implemented which achieve the same functionality. In a conceptual sense, any arrangement of components to achieve the same functionality is effectively "associated" such that the desired functionality is achieved. Hence, any two components herein combined to achieve a particular functionality can be seen as "associated with" each other such that the desired functionality is achieved, irrespective of architectures or intermedial components. Likewise, any two components so associated can also be viewed as being "operably connected", or "operably coupled", to each other to achieve the desired functionality.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that, based upon the teachings herein, changes and modifications may be made without departing from this invention and its broader aspects and, therefore, the appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims. It will be understood by those within the art that if a specific number of an introduced claim element is intended, such an intent will be explicitly recited in the claim, and in the absence of such recitation no such intent is present. For example, as an aid to understanding, the following appended claims may contain usage of the introductory phrases "at least one" and "one or more" to introduce claim elements. However, the use of such phrases should <u>not</u> be construed to imply that the introduction of a claim element by the indefinite articles "a" or "an" limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an"; the same holds true for the use of definite articles used to introduce claim elements. In addition, even if a specific number of an introduced claim element *is* explicitly recited, those skilled in the art will recognize that such recitation should typically be interpreted to mean *at least* the recited number (e.g., the bare recitation of "two elements," without other modifiers, typically means *at least* two elements, or *two or more* elements).

WHAT IS CLAIMED IS:

1. A method comprising:
capturing an image actually displayed via a display sighting system;
computing at least one difference between the captured image and a recalled representation of the image theoretically displayed via the display sighting system; and
presenting the computed at least one difference via a visual display device.

2. The method of Claim 1, wherein said computing at least one difference between the captured image and a recalled representation of the image theoretically displayed via the display sighting system further comprises:
computing at least one angular difference between an angular orientation of the captured image and the recalled representation of the image theoretically displayed via the display sighting system;
computing at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system; or
computing at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system.

3. The method of Claim 2, wherein the recalled representation of the image theoretically displayed via the display sighting system comprises a test pattern having at least one vertical line.

4. The method of Claim 2, wherein the recalled representation of the image theoretically displayed via the display sighting system comprises a test pattern having at least one horizontal line.

5. The method of Claim 2, wherein said presenting the computed at least one difference via a visual display device further comprises:

presenting the at least one angular difference between an angular orientation of the captured image and the recalled representation of the image theoretically displayed via the display sighting system;

presenting the at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system; or

presenting the at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system.

6. The method of Claim 1, wherein said computing at least one difference between the captured image and a recalled representation of the image theoretically displayed via the display sighting system further comprises:

computing at least one gray-shades-displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system;

computing at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system; or

computing at least one predicted focus magnitude indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system.

7. The method of Claim 6, wherein the recalled representation of the image theoretically displayed via the display sighting system comprises a test pattern having at least two gray shades.

8. The method of Claim 6, wherein said presenting the computed at least one difference via a visual display device further comprises:

presenting the at least one gray-shades-displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system;

presenting the at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system; or

presenting the at least one predicted focus magnitude indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system.

9. The method of Claim 1, wherein said capturing an image actually displayed via a display sighting system further comprises:

capturing the image via a camera.

10. The method of Claim 9, wherein said capturing the image via a camera further comprises:

capturing the image via a data acquisition card interposed between a narrow-angle camera and a portable computer system; or

capturing the image via a data acquisition card interposed between a wide-angle camera and the portable computer system.

11. A system comprising:

circuitry for capturing an image actually displayed via a display sighting system, wherein said circuitry for capturing an image includes one or more electrical circuits selected from the group including but not limited to electrical circuits having at least one discrete electrical circuit, electrical circuits having at least one integrated circuit,

electrical circuits having at least one application specific integrated circuit, and electrical circuits providing at least one general purpose computing device configurable by a computer program;

circuitry for computing at least one difference between the captured image and a recalled representation of the image theoretically displayed via the display sighting system, wherein said circuitry for computing includes one or more electrical circuits selected from the group including but not limited to electrical circuits having at least one discrete electrical circuit, electrical circuits having at least one integrated circuit, electrical circuits having at least one application specific integrated circuit, and electrical circuits providing at least one general purpose computing device configurable by a computer program; and

circuitry for presenting the computed at least one difference via a visual display device, wherein said circuitry for presenting includes one or more electrical circuits selected from the group including but not limited to electrical circuits having at least one discrete electrical circuit, electrical circuits having at least one integrated circuit, electrical circuits having at least one application specific integrated circuit, and electrical circuits providing at least one general purpose computing device configurable by a computer program.

12. The system of Claim 11, wherein said circuitry for computing at least one difference between the captured image and a recalled representation of the image theoretically displayed via the display sighting system further comprises:

circuitry for computing at least one angular difference between an angular orientation of the captured image and the recalled representation of the image theoretically displayed via the display sighting system;

circuitry for computing at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system; or

circuitry for computing at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system.

13. The system of Claim 12, wherein the recalled representation of the image theoretically displayed via the display sighting system comprises a test pattern having at least one vertical line.

14. The system of Claim 12, wherein the recalled representation of the image theoretically displayed via the display sighting system comprises a test pattern having at least one horizontal line.

15. The system of Claim 12, wherein said circuitry for presenting the computed at least one difference via a visual display device further comprises:

circuitry for presenting the at least one angular difference between an angular orientation of the captured image and the recalled representation of the image theoretically displayed via the display sighting system;

circuitry for presenting the at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system; or

circuitry for presenting the at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system.

16. The system of Claim 11, wherein said circuitry for computing at least one difference between the captured image and a recalled representation of the image theoretically displayed via the display sighting system further comprises:

circuitry for computing at least one gray-shades-displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system;

circuitry for computing at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system; or

circuitry for computing at least one predicted focus magnitude indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system.

17. The system of Claim 16, wherein the recalled representation of the image theoretically displayed via the display sighting system comprises a test pattern having at least two gray shades.

18. The system of Claim 16, wherein said circuitry for presenting the computed at least one difference via a visual display device further comprises:

circuitry for presenting the at least one gray-shades-displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system;

circuitry for presenting the at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system; or

circuitry for presenting the at least one predicted focus magnitude indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system.

19. The system of Claim 11, wherein said circuitry for capturing an image actually displayed via a display sighting system further comprises:

circuitry for capturing the image via a camera.

20. The system of Claim 19, wherein said circuitry for capturing the image via a camera further comprises:

circuitry for capturing the image via a data acquisition card interposed between a narrow-angle camera and a portable computer system; or

circuitry for capturing the image via a data acquisition card interposed between a wide-angle camera and the portable computer system.

21. An image capturing device comprising:
a Helmet Display Unit (HDU) holding fixture; and
at least one camera mounted proximate to the HDU holding fixture.

22. The image capturing device of Claim 21, wherein the Helmet Display Unit (HDU) holding fixture comprises:
the Helmet Display Unit (HDU) holding fixture movable between at least two positions.

23. The image capturing device of Claim 21, wherein the Helmet Display Unit (HDU) holding fixture comprises:
the Helmet Display Unit (HDU) holding fixture is attached to a lever-spring assembly.

24. The image capturing device of Claim 21, wherein said at least one camera mounted proximate to the HDU holding fixture comprises:
at least one wide-angle and at least one narrow-angle camera.

IMAGE QUALITY TESTER

Clarence E. Rash
Sheng-Jen Hsieh
Thomas H. Harding
Howard H. Beasley
John S. Martin
Ronald W. Reynolds
Robert M. Dillard

ABSTRACT OF THE DISCLOSURE

A method including but not limited to capturing an image actually displayed via a display sighting system; computing at least one difference between the captured image and a recalled representation of the image theoretically displayed via the display sighting system; and presenting the computed at least one difference via a visual display device. In various implementations, circuitry is used to effect the foregoing-described method; the circuitry can be virtually any combination of hardware, software, and/or firmware configured to effect the foregoing-described method depending upon the design choices of the system designer.

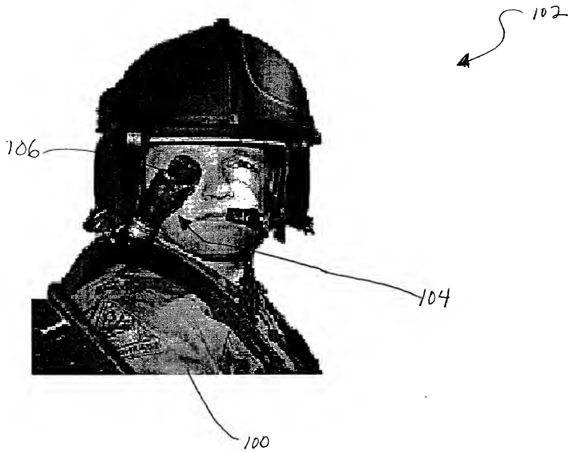


FIG. 1

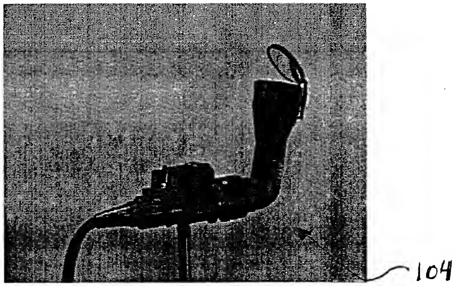


FIG. 2

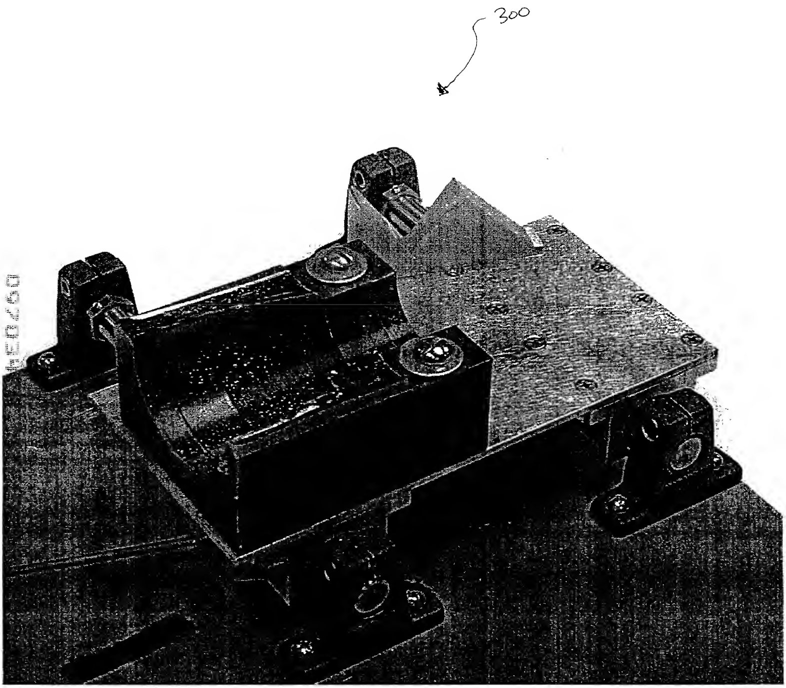


FIG 3A

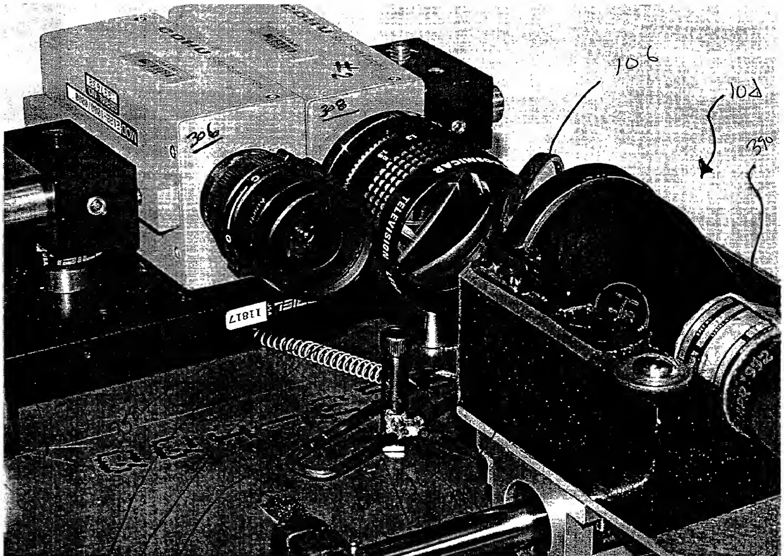


FIG. 3B

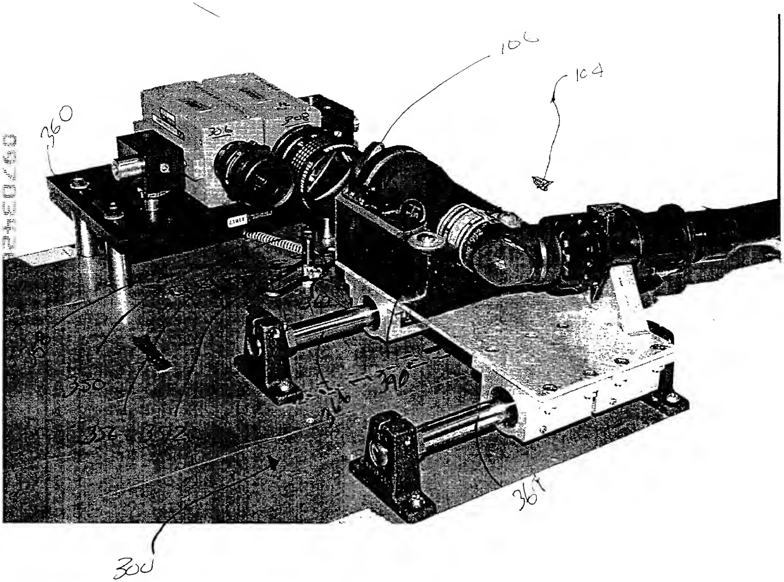


FIG 3C

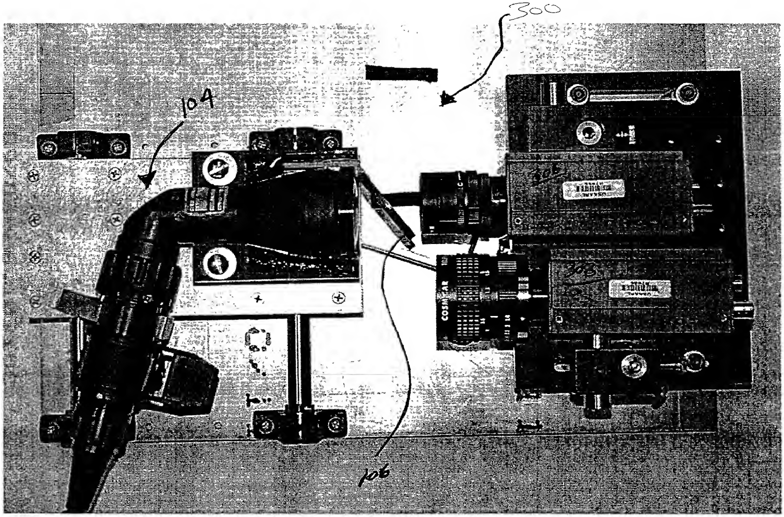


FIG. 3D

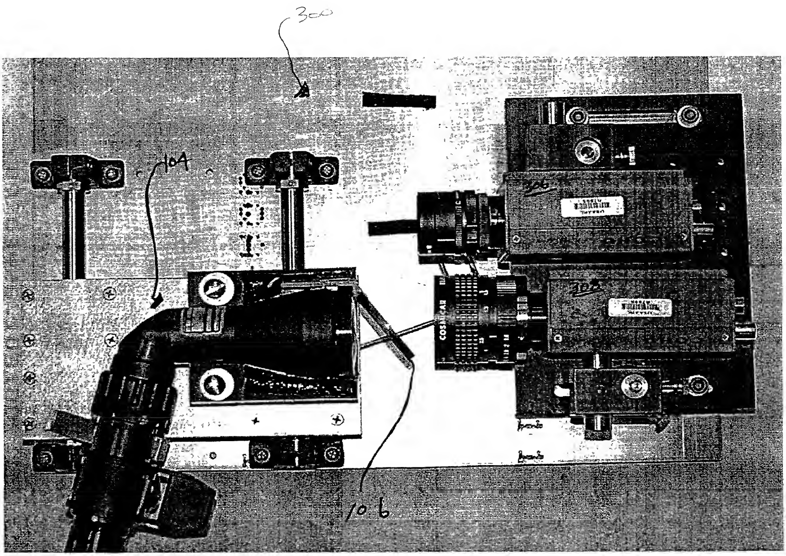


FIG 3R

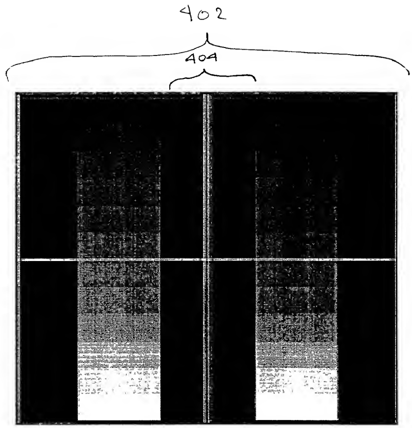
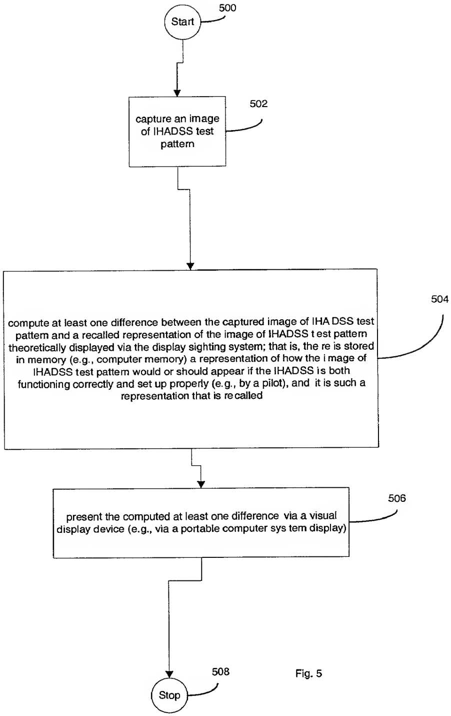


FIG. 4



Start
500
capture an image of IHADSS test pattern
502
compute at least one difference between the captured image of IHA DSS test pattern and a recalled representation of the image of IHADSS t est pattern theoretically displayed via the display sighting system; that is, the re is stored in memory (e.g., computer memory) a representation of how the i mage of IHADSS test pattern would or should appear if the IHADSS is both functioning correctly and set up properly (e.g., by a pilot), and it is such a representation that is re called
504
present the computed at least one difference via a visual display device (e.g., via a portable computer sys tem display)
506
Stop
508

Fig. 5

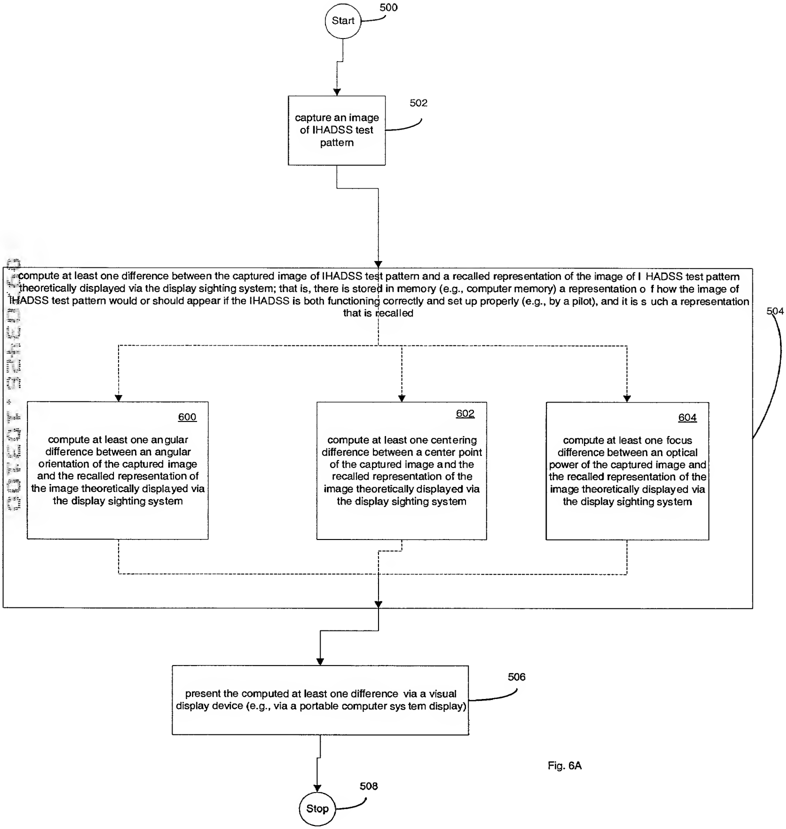

Start
500
capture an image of IHADSS test pattern
502
compute at least one difference between the captured image of IHADSS test pattern and a recalled representation of the image of I HADSS test pattern theoretically displayed via the display sighting system; that is, there is stored in memory (e.g., computer memory) a representation o f how the image of IHADSS test pattern would or should appear if the IHADSS is both functioning correctly and set up properly (e.g., by a pilot), and it is s uch a representation that is recalled
504
600
compute at least one angular difference between an angular orientation of the captured image and the recalled representation of the image theoretically displayed via the display sighting system
602
compute at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system
604
compute at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system
present the computed at least one difference via a visual display device (e.g., via a portable computer sys tem display)
506
Stop
508

Fig. 6A

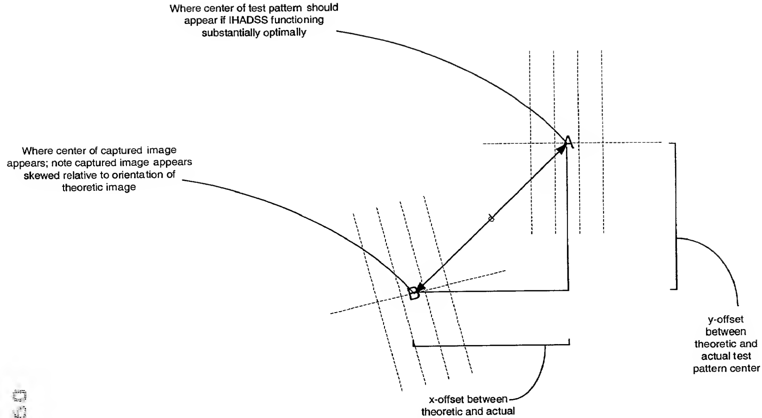

Where center of test pattern should appear if IHADSS functioning substantially optimally
A
Where center of captured image appears; note captured image appears skewed relative to orientation of theoretic image
B
y-offset between theoretic and actual test pattern center
x-offset between theoretic and actual test pattern center

Fig. 6B

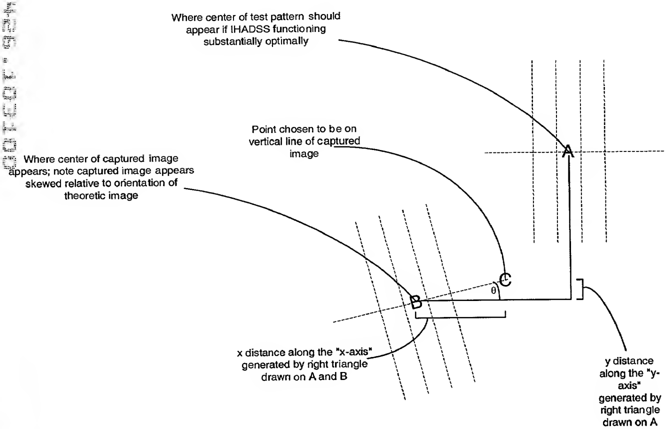

Where center of test pattern should appear if IHADSS functioning substantially optimally
Point chosen to be on vertical line of captured image
A
Where center of captured image appears; note captured image appears skewed relative to orientation of theoretic image
C
θ
B
x distance along the "x-axis" generated by right triangle drawn on A and B
y distance along the "y-axis" generated by right triangle drawn on A and B

Fig. 6C

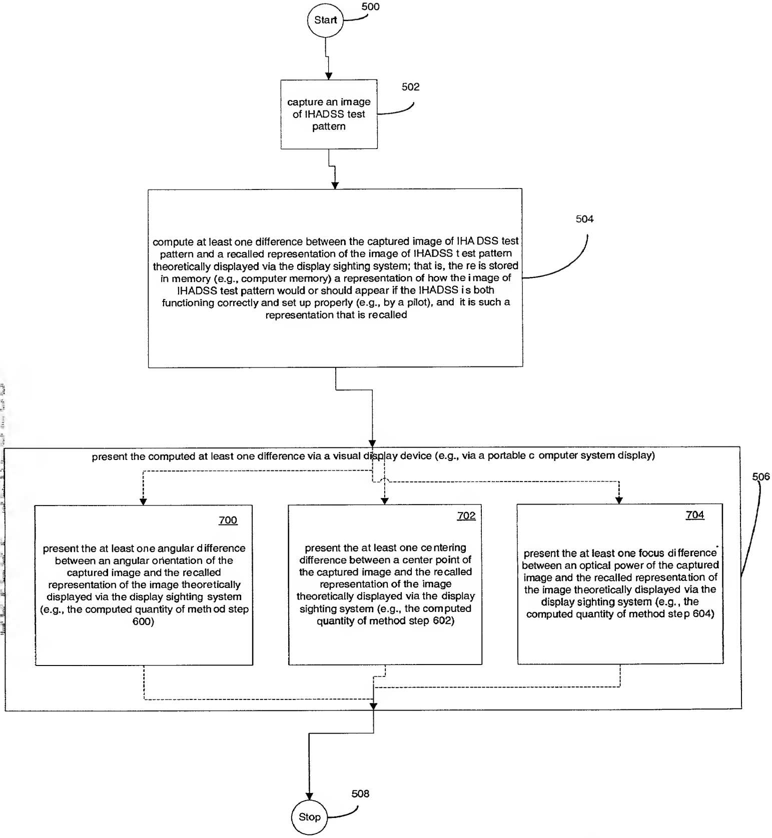
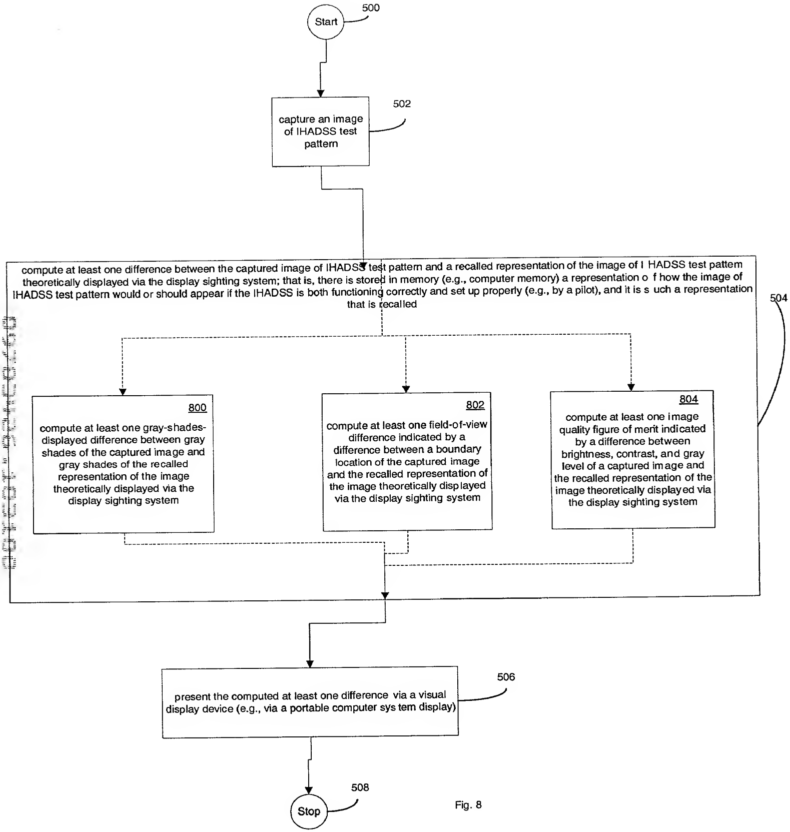

500
Start
502
capture an image of IHADSS test pattern
504
compute at least one difference between the captured image of IHADSS test pattern and a recalled representation of the image of IHADSS test pattern theoretically displayed via the display sighting system; that is, there is stored in memory (e.g., computer memory) a representation of how the image of IHADSS test pattern would or should appear if the IHADSS is both functioning correctly and set up properly (e.g., by a pilot), and it is such a representation that is recalled
506
present the computed at least one difference via a visual display device (e.g., via a portable computer system display)
700
present the at least one angular difference between an angular orientation of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 600)
702
present the at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 602)
704
present the at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 604)
508
Stop

Fig. 7



500
Start
502
capture an image of IHADSS test pattern
compute at least one difference between the captured image of IHADSS test pattern and a recalled representation of the image of I HADSS test pattern theoretically displayed via the display sighting system; that is, there is stored in memory (e.g., computer memory) a representation o f how the image of IHADSS test pattern would or should appear if the IHADSS is both functioning correctly and set up properly (e.g., by a pilot), and it is s uch a representation that is recalled
504
800
compute at least one gray-shades-displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system
802
compute at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system
804
compute at least one image quality figure of merit indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system
506
present the computed at least one difference via a visual display device (e.g., via a portable computer sys tem display)
508
Stop

Fig. 8

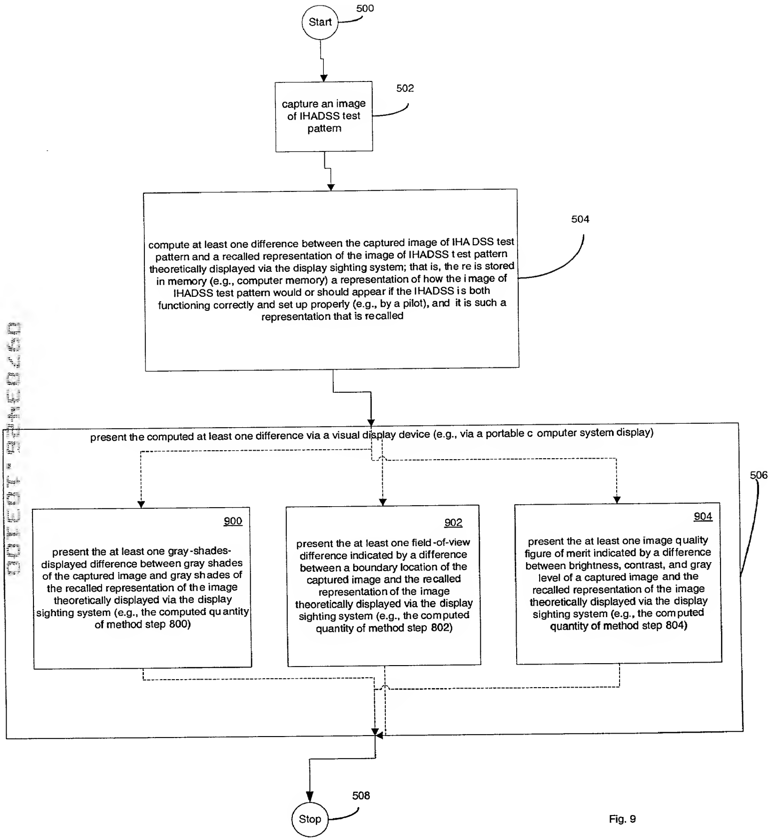

Start
500
capture an image of IHADSS test pattern
502
compute at least one difference between the captured image of IHADSS test pattern and a recalled representation of the image of IHADSS test pattern theoretically displayed via the display sighting system; that is, there is stored in memory (e.g., computer memory) a representation of how the image of IHADSS test pattern would or should appear if the IHADSS is both functioning correctly and set up properly (e.g., by a pilot), and it is such a representation that is recalled
504
present the computed at least one difference via a visual display device (e.g., via a portable computer system display)
506
900
present the at least one gray-shades-displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 800)
902
present the at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 802)
904
present the at least one image quality figure of merit indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 804)
Stop
508

Fig. 9

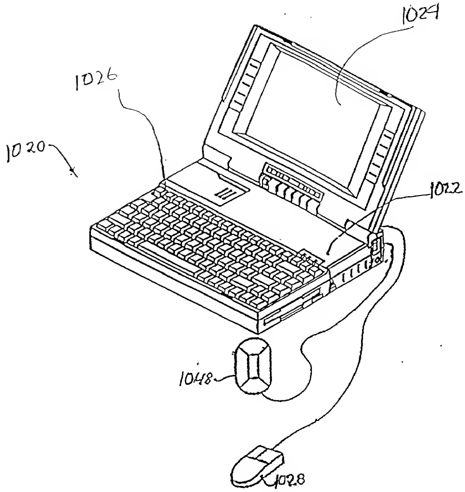


FIG 10

Figure A1. The IHADSS

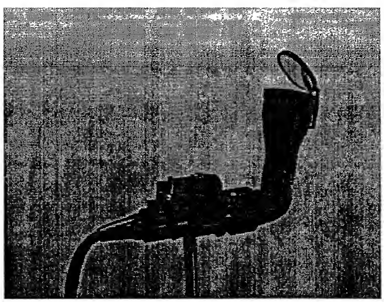


Figure A2. The IHADSS HDU.

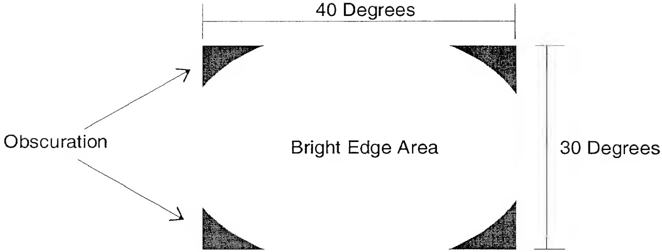


Figure A3. Display size.

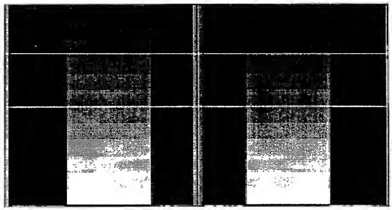


Figure A4. IHADSS built-in test pattern.

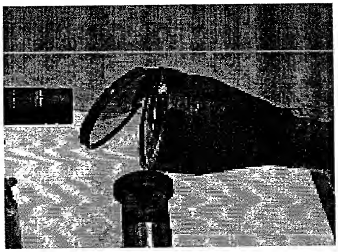


Figure A5. Previous design.

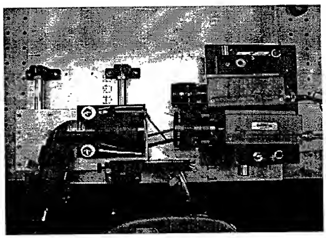


Figure A6. Revised design.

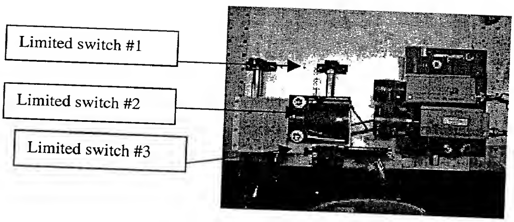


Figure A7. Locations of sensors in proposed HMD fixture design.

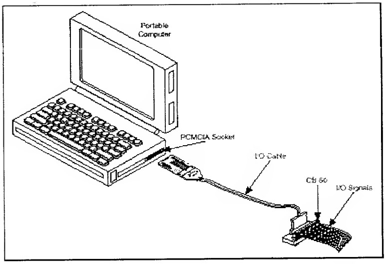


Figure A8. Typical DAQCard-DIO-24 configuration.

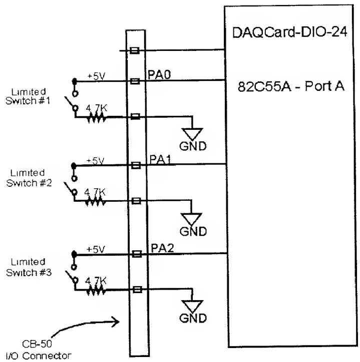


Figure A9. Schematic diagram of proposed design.

1.

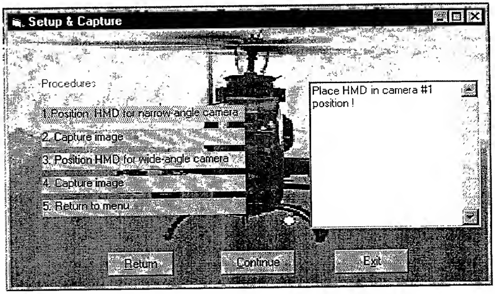


Figure A10. Initial display screen, switches open.

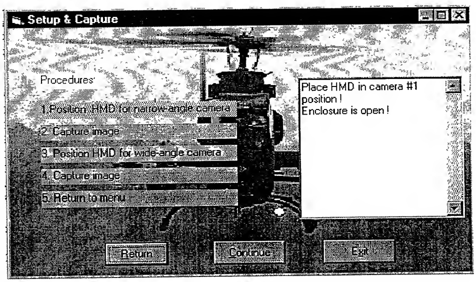


Figure A11. Display screen, Continue button, switches open.

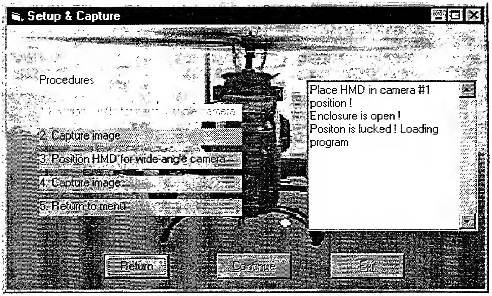


Figure A12. Display screen switches 1 and 2 pressed.

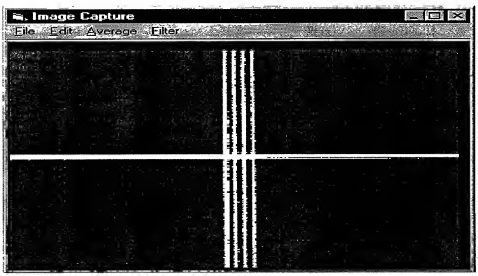


Figure A13. Display screen, image capture
module activated.

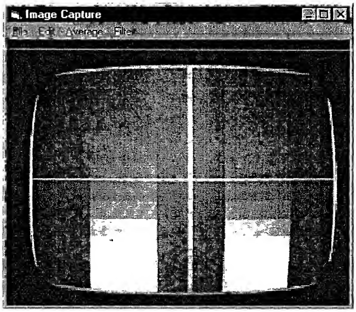


Figure A14. Screenshot of image capture module.

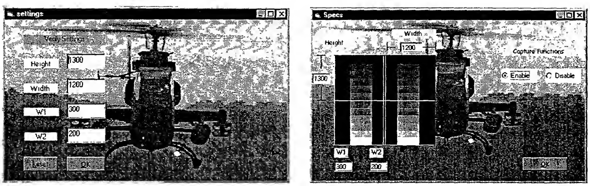


Figure A15. Screenshots of parameter setting display screens.

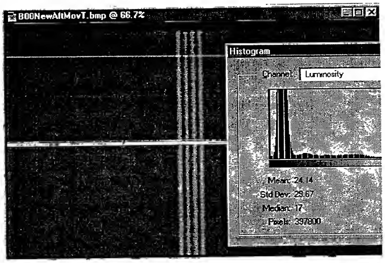


Figure A16. Original image.

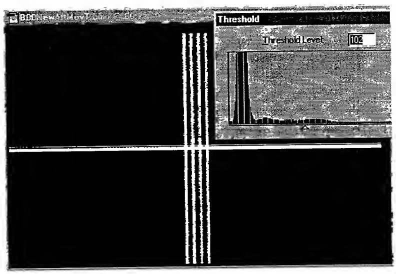


Figure A17. Image after binary processing.

Figure B1. The IHU of the AH-64 IHADSS.

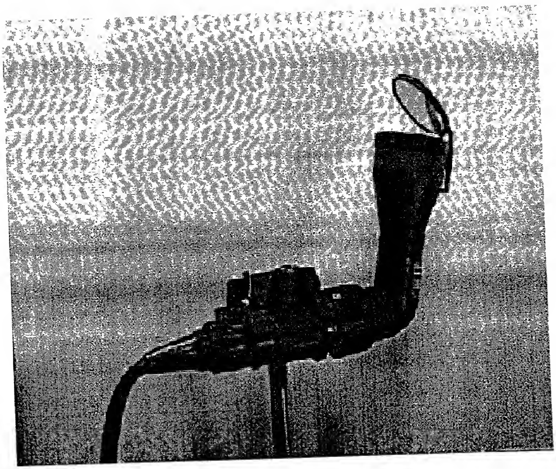


Figure B2. The IHADSS HDU.

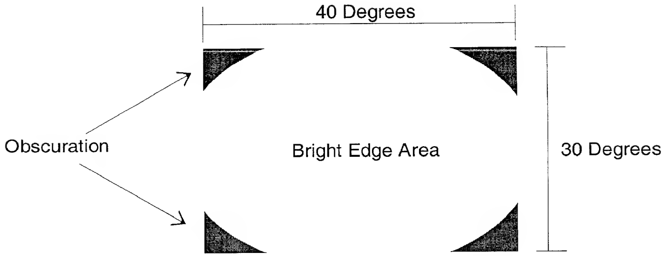


Figure B3. Display size.

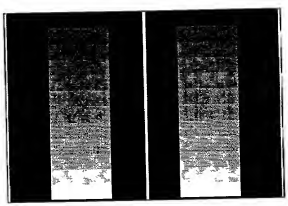


Figure B4. Test pattern from the IHADSS HMD.

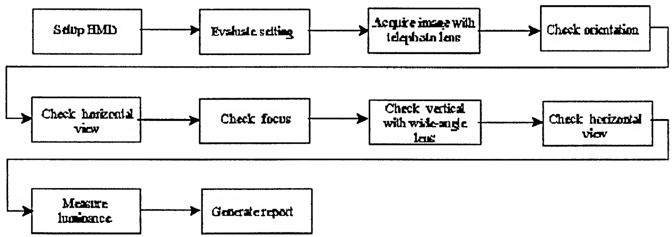


Figure B5. Flow chart for HMD prototype tester operation.

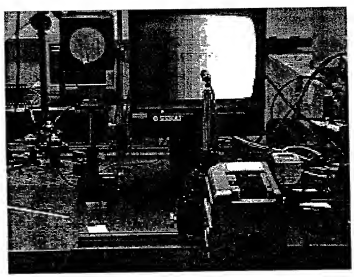


Figure B6. Experimental setup for camera sensitivity analysis.

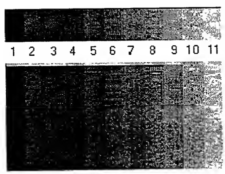


Figure B7. Sampling locations on the test pattern.

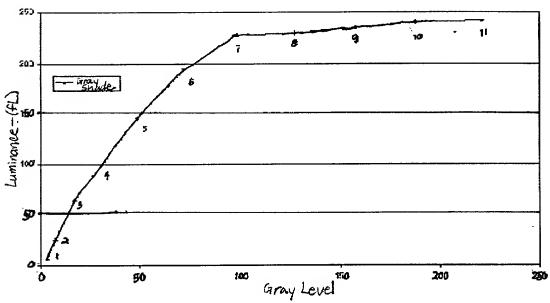


Figure B8. Plot of photometer and CCD Camera data.

Figure B9. Set up for test pattern measurement.

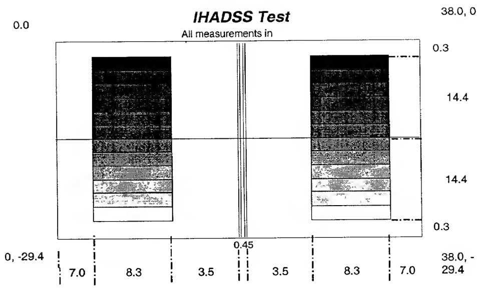


Figure B10. Test pattern design based on measurement results.

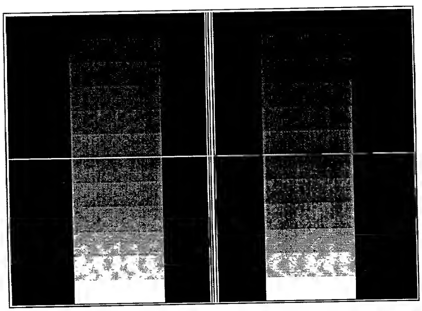


Figure B11. Replicated test pattern image.

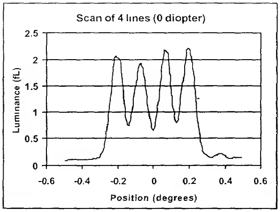


Figure B12. Measurement of luminance of the center lines.

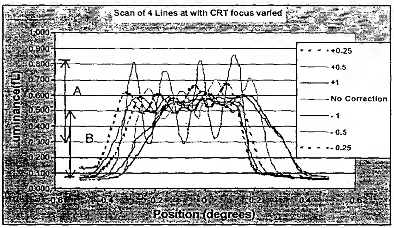


Figure B13. Center lines measurement with varied focus.

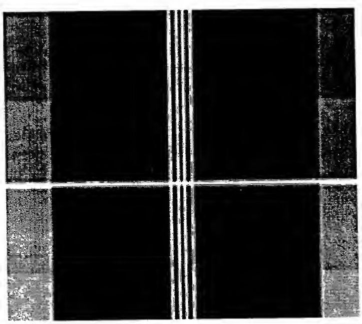


Figure B14. Designed test pattern with focus on the center lines.

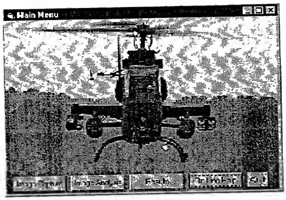


Figure B15. Opening screen of prototype software.

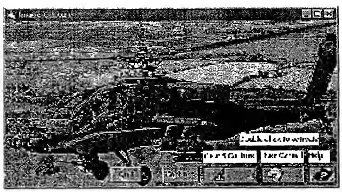


Figure B16. Image capture module.

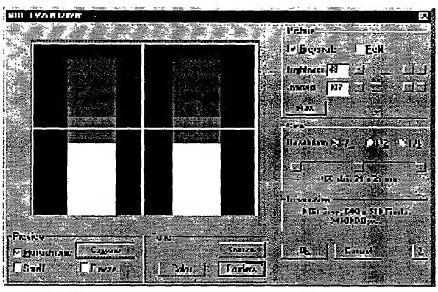


Figure B17. Image Capture Component.

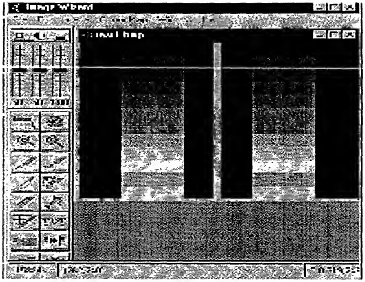


Figure B18. Image processing component.

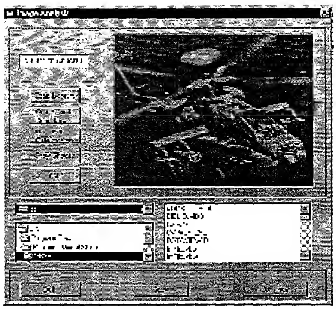


Figure 69. Image analysis and interpretation module.

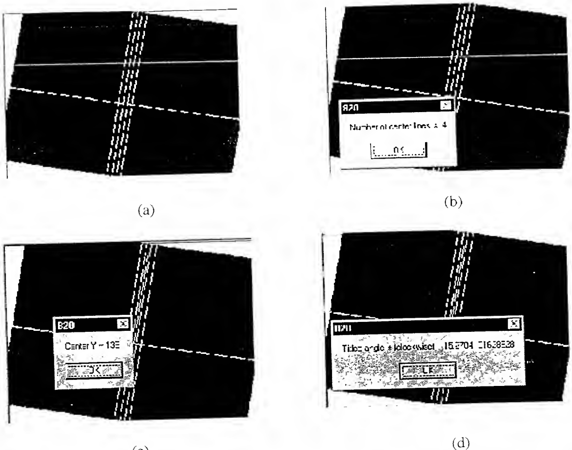


Figure B20. Tilted test pattern binary images from image analysis module.

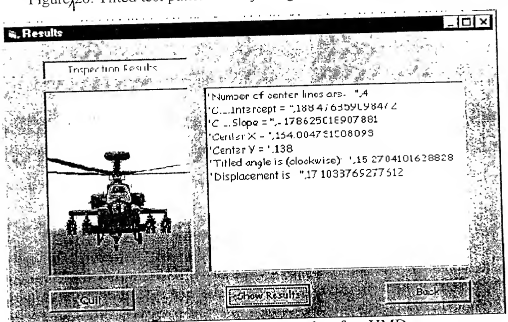


Figure B21. Overall testing results of an HMD.

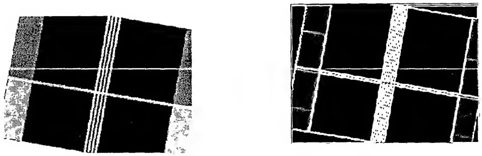


Figure B22. Tilted test pattern before (left) and after (right) Sober edge detection.

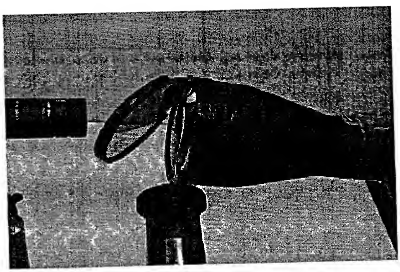
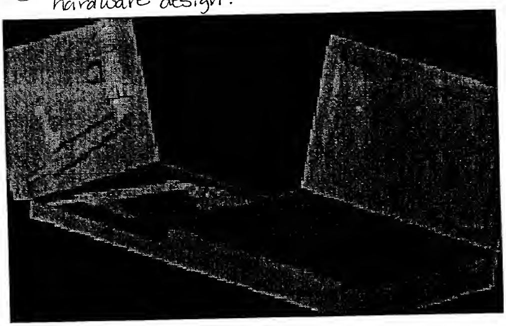


Figure B23. Investigation of CCD image capture arrangement.

Figure B24. CAD concept of prototype hardware design.



DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below adjacent to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of subject matter (process, machine, manufacture, or composition of matter, or an improvement thereof) which is claimed and for which a patent is sought by way of the application entitled

Image Quality Tester

which (check) ☒ is attached hereto.
☐ and is amended by the Preliminary Amendment attached hereto.
☐ was filed on as Application Serial No.
☐ and was amended on (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information, which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, § 119(a)-(d) of any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
Number	Country	Day/Month/Year Filed	Yes	No
N/A			☐	☐

I hereby claim the benefit under Title 35, United States Code, § 119(e) of any United States provisional application(s) listed below:

Provisional Application Number	Filing Date
Not yet assigned	October 11, 2000

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) or PCT international application(s) designating the United States of America listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior application(s) in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose information, which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56, which became available between the filing date of the prior application(s) and the national or PCT international filing date of this application:

Application Serial No.	Filing Date	Status (patented, pending, abandoned)
N/A		

I hereby appoint the following practitioners to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith:

Customer Number 24251

Please address all correspondence and telephone calls to:

Dale R. Cook
SKJERVEN MORRILL MacPHERSON LLP
25 Metro Drive, Suite 700
San Jose, California 95110-1349

Telephone: 408-453-9200
Facsimile: 408-453-7979

I declare that all statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and all statements made herein are made with the knowledge that whoever, in any matter within the jurisdiction of the Patent and Trademark Office, knowingly and willfully falsifies, conceals, or covers up by any trick, scheme, or device a material fact, or makes any false, fictitious or fraudulent statements or representations, or makes or uses any false writing or document knowing the same to contain any false, fictitious or fraudulent statement or entry, shall be subject to the penalties including fine or imprisonment or both as set forth under 18 U.S.C. 1001, and that violations of this paragraph may jeopardize the validity of the application or this document, or the validity or enforceability of any patent, trademark registration, or certificate resulting therefrom.

Full name of sole (or first joint) inventor:	Clarence E. Rash		
Inventor's Signature:		Date:	
Residence:	Enterprise, Alabama		
Post Office Address:	110 Victoria Drive Enterprise, Alabama 36330	Citizenship:	United States

Full name of sole (or second joint) inventor:	Thomas H. Harding		
Inventor's Signature:		Date:	
Residence:	Enterprise, Alabama		
Post Office Address:	406 Lake Oliver Drive Enterprise, Alabama 36330	Citizenship:	United States

Full name of sole (or third joint) inventor:	Sheng-Jen Hsieh		
Inventor's Signature:		Date:	
Residence:	College Station, Texas		
Post Office Address:	1703-A Anderson Street College Station, Texas 77840	Citizenship:	United States

Full name of sole (or fourth joint) inventor: Howard H. Beasley
Inventor's Signature: ______________________ Date: ______________
Residence: Louisville, Alabama
Post Office Address: 1587 Highway 51 South
Louisville, Alabama 36048
Citizenship: United States

Full name of sole (or fifth joint) inventor: John S. Martin
Inventor's Signature: ______________________ Date: ______________
Residence: Daleville, Alabama
Post Office Address: 15 Woodland Court
Daleville, Alabama 36322
Citizenship: United States

Full name of sole (or sixth joint) inventor: Ronald W. Reynolds
Inventor's Signature: ______________________ Date: ______________
Residence: Geneva County, Hwy, 45 North,
Bellwood, Alabama
Post Office Address: P.O. Box 56
Bellwood, Alabama 36313
Citizenship: United States

Full name of sole (or seventh joint) inventor: Robert M. Dillard
Inventor's Signature: ______________________ Date: ______________
Residence: Ozark, Alabama
Post Office Address: Rural Route 5, Box 90
Ozark, Alabama 36360
Citizenship: United States

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

SCANNED # 6

Application deficiencies were found during scanning:

☐ Page(s)__________ of ______________________________ were not present for scanning.
(Document title)

☐ Page(s)__________ of ______________________________ were not present for scanning.
(Document title)

☒ Scanned copy is best available. drawings